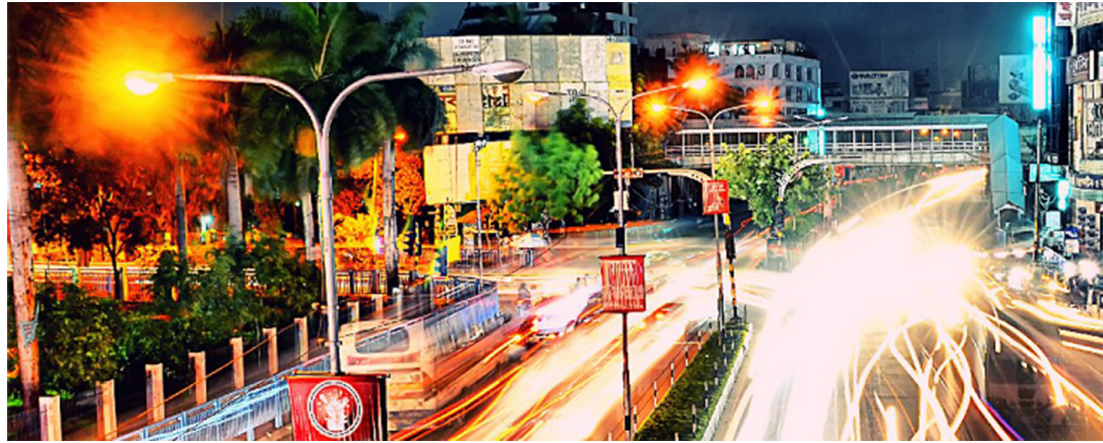


PayPal Express Checkout Using Alternative Payment Services

Simple Order API

Developer Guide



© 2022. 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 Unpublished rights reserved under the copyright laws of the United States.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, the Cybersource logo, and 3-D Secure 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: 22.02

Contents

- Recent Revisions to This Document..... 6**
- About This Guide..... 7**
- Introduction to PayPal Express Checkout..... 9**
 - Benefits of Using PayPal Express Checkout..... 9
 - Prerequisites..... 10
 - PayPal Merchant Set Up..... 10
 - Acquiring and Providing PayPal Merchant Set Up Information to Cybersource..... 10
 - PayPal Express Checkout In-Context Flow..... 11
 - PayPal Express Order Processing..... 12
 - Using Standard Orders..... 12
 - Using Custom Orders..... 12
 - PayPal Express Order Tracking..... 13
 - Reconciliation IDs..... 14
 - Using PayPal Express Credit..... 14
 - Adding the PayPal Express Credit Banner and Button to Your Page..... 15
 - Additional PayPal Express Checkout Services..... 15
 - Obtaining Transaction Information..... 16
 - Dispute and Chargeback Services..... 16
- Sessions Service..... 18**
 - Requesting the Sessions Service..... 18
 - Following the Sessions Request..... 18
- Check Status Service..... 20**
 - Requesting the Check Status Service..... 20
 - Checking the Status of a Service Request..... 20
 - Checking the Status of an Individual Service Request..... 21
- Order Service..... 24**
 - Requesting the Order Service..... 24
 - Following the Order Request..... 24
- Cancel Service..... 25**
 - Requesting the Cancel Service..... 25
- Authorization Service..... 26**
 - Requesting the Authorization Service..... 26

Following the Authorization Request.....	26
Authorization Reversal Service.....	27
Requesting the Authorization Reversal Service.....	27
Capture Service.....	28
Requesting the Capture Service.....	28
Following the Capture Request.....	28
Sale Service.....	29
Requesting the Sale Service.....	29
Following the Sale Request.....	29
Refund Service.....	30
Requesting the Refund Service.....	30
Billing Agreements Service.....	31
Requesting the Billing Agreements Service.....	31
Setting Up a Billing Agreement.....	31
Creating a Billing Agreement.....	31
Billing Agreement Service Examples.....	32
API Fields.....	34
Formatting Restrictions.....	34
Data Type Definitions.....	34
Numbered Elements.....	34
Working with Multibyte Character Strings.....	35
Request IDs in Request Messages.....	36
Request IDs in API Response Messages.....	37
Request Fields.....	38
Response Fields.....	52
Service Examples.....	73
Name-Value Pair Examples.....	73
Sessions Service.....	73
Order Service.....	74
Cancel Service.....	75
Authorization Service.....	76
Authorization Reversal Service.....	76
Capture Service.....	77
Sale Service.....	77
Refund Service.....	79

Billing Agreement Service.....	80
XML Examples.....	81
Sessions Service.....	81
Order Service.....	82
Cancel Service.....	84
Authorization Service.....	85
Authorization Reversal Service.....	85
Capture Service.....	86
Sale Service.....	87
Refund Service.....	90
Billing Agreement Service.....	90
Check Status Examples.....	92
Name-Value Pair Examples.....	92
Sessions Service: Check Status.....	92
Order Service: Check Status.....	93
Authorization Service: Check Status.....	93
Capture Service: Check Status.....	94
Sale Service: Check Status.....	94
Refund Service: Check Status.....	95
Billing Agreement Service: Check Status.....	95
XML Examples.....	96
Sessions Service: Check Status.....	96
Order Service: Check Status.....	97
Authorization Service: Check Status.....	98
Capture Service: Check Status.....	99
Sale Service: Check Status.....	99
Refund Service: Check Status.....	100
Billing Agreement Service: Check Status.....	101
Reason Codes.....	103

Recent Revisions to This Document

Release	Changes
22.02	This revision contains only editorial changes and no technical updates.
22.01	This revision contains only editorial changes and no technical updates.
21.01	Updated description of field shipTo_immutable in API Fields (on page 34) .
20.01	Added sale service request and response examples for a reference transaction to Service Examples (on page 73). Added apCheckStatusService_checkStatusRequestID field to API Fields (on page 34) and Service Examples (on page 73).
19.04	Added name-value pair examples for the Simple Order API to Service Examples (on page 73) and Check Status Examples (on page 92). Added information on Order Tracking to PayPal Express Order Tracking (on page 13). Added shipTo_shippingMethod field to API Fields (on page 34).
19.03	Added updated information about PayPal customer billing addresses and check status service response fields in API Fields (on page 34) .
19.02	Reorganized document structure: • Merged content to create a single section: Check Status Service (on page 20). • Moved relevant Simple Order API information to API Fields (on page 34). • Consolidated service examples in Service Examples (on page 73). • Consolidated check status examples in Check Status Examples (on page 92). Added new clientMetadataID field to API Fields (on page 34).

About This Guide

This section describes the audience and purpose of this guide as well as conventions and related documentation. See below information about how to use this guide and where to find further information.

Audience and Purpose

This guide is written for application developers who want to use the Cybersource Simple Order API to integrate PayPal Express Checkout into their order management system.

Implementing PayPal Express Checkout services requires software development skills. You must write code that uses API request and response fields to integrate PayPal Express Checkout services.

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

- *Getting Started with Cybersource Advanced for the Simple Order API* ([PDF](#) | [HTML](#)) describes how to get started using the Simple Order API.
- [Simple Order API and SOAP Toolkit API Testing Information page](#)

Refer to the Support Center for complete technical documentation:

<https://www.cybersource.com/en-us/support/technical-documentation.html>

Customer Support

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

<http://www.cybersource.com/support>

Introduction to PayPal Express Checkout

When you use the PayPal Express Checkout services:

- No additional security key management is required from you.
- You manage your PayPal Express Checkout payments in the Cybersource Business Center as you would other payment types. You can search for the payment and view its details.
- You use the Transaction Details Report and the Transaction Summary Report to manage your PayPal Express Checkout payments.

Benefits of Using PayPal Express Checkout

When you use PayPal Express Checkout, your customer completes the order on your web site rather than on the PayPal web site. You can:

- Receive real-time notification of successful payments.
- Enable your customers to be directed to your web site's final confirmation page.
- Receive notification from Cybersource that the customer's address has been confirmed and that you are eligible for coverage under the PayPal Seller Protection Policy (if you are enrolled). For more information, see [Dispute and Chargeback Services \(on page 16\)](#).

PayPal requires you to provide three ways for your customers to access PayPal Express Checkout. See [Table 2: PayPal Express Checkout Methods \(on page 9\)](#) below.

PayPal Express Checkout Methods

Method	Description
Shortcut	Place the Checkout with PayPal button on your checkout page.
Payment	Place the PayPal button on your payment page or add PayPal as a choice in your payment drop-down list.
Credit	Place the PayPal Credit button on your checkout page.

Prerequisites

Before you process PayPal Express Checkout payments:

- Install a Cybersource client. See:

Getting Started with Cybersource Advanced for the Simple Order API ([PDF](#) | [HTML](#))

- Contact Customer Support to set up your Cybersource merchant account. For more information, see [PayPal Merchant Set Up \(on page 10\)](#).
- If you are using the Simple Order API with XML, make sure that you are using the correct API version for the features that you are implementing. Cybersource recommends that you use the most recent API version.

For general information about the API versions, see [Getting Started with Cybersource Advanced for the Simple Order API](#).

PayPal Merchant Set Up

To process your PayPal Express Checkout payments through Cybersource, you must set up your merchant account by sharing this information with Cybersource:

- PayPal sandbox account (facilitator email address)
- PayPal client ID (CID)
- PayPal secret
- PayPal merchant ID (MID)

Acquiring and Providing PayPal Merchant Set Up Information to Cybersource

1. If you do not already have one, set up a PayPal business or premier account:
<https://www.paypal.com/us/webapps/mpp/merchant>
2. If you do not already have one, set up a PayPal developer sandbox account:
<https://developer.paypal.com/docs/classic/lifecycle/sandbox/accounts/#create-and-manage-sandbox-accounts>
3. As directed on the PayPal site, get your PayPal credentials by creating a PayPal app:
<https://developer.paypal.com/docs/api/overview/#get-credentials>

4. Save the Sandbox account facilitator email address, the client ID, and the secret for future reference. For example:

- Sandbox account facilitator email address:

`merchantuser-facilitator@merchant.com`

- Client ID:

`AahnQzKLL2vvG_UI6YQy9xcyt5joMLVoPHW-1Bv8gCvPkTiNwQSRcvKIKXy8UZZguijbwJTTs_Cjhdz`

- Secret:

`EOE3eqqeIBy4q8LhsON0-wp2zPb_0SOqPH3sopx_uwuIMkCug7zw3aKDunstrXmcrGecmpeUJgsqTGO`

5. On the PayPal developer site, click **Profile** at the top right corner of the page.

6. Navigate to **My business info** in the left navigation pane. Your merchant ID is displayed in the list of merchant information. Save the merchant ID for future reference. For example:

- Merchant ID: `PP45KFYYH69KU`

7. Contact Cybersource merchant support and provide them with the information you have collected from PayPal: the sandbox facilitator email account, the client ID, the secret, and the merchant ID.

When your Cybersource account is set up, you can process PayPal Express Checkout payments.

PayPal Express Checkout In-Context Flow

 **Important:** To enable PayPal Express Checkout, see:

<https://developer.paypal.com/docs/checkout/reference/upgrade-integration/>

When your customer is ready to pay for the products that they choose from your web site:

1. The customer clicks **Checkout with PayPal** on your checkout page or clicks **PayPal** on your payment page.
2. Your web site remains in view while a PayPal window appears.
3. The customer logs in and chooses a payment method and shipping address.
4. Customer confirms payment, and PayPal redirects the customer's browser to your web site. You can choose to display a payment confirmation page:

PayPal Express Order Processing

PayPal Express checkout provides two kinds of order processing:

- Standard orders
- Custom orders

The kind of order processing you use depends upon your business. Cybersource recommends using custom orders, because they can be more flexible than standard orders.

Using Standard Orders

Use a standard order when you meet one or more of these criteria:

- You do not usually run out of products.
- You do not usually have back orders.
- You do not accept advanced orders.

Standard Order Flow

Sequence of service requests for a standard 100.00 order:

1. Sessions Service 100.00
2. Check Status Service
3. Order Service for 100.00
4. Sale Service for 100.00

Using Custom Orders

A PayPal Express checkout custom order enables you to perform multiple authorizations and multiple captures for each authorization. Use a custom order when you meet one or more of these criteria:

- You often have split shipments.
- You accept advanced orders for products that are not currently available but will be available within the next 29 days.
- You often have back orders for which the customer waits longer than a week.
- You often reauthorize bank cards.

Custom Order Flows

Sequence of Service Requests for Custom 100.00 Orders

Immediate Sale	Delayed Authorization	Delayed Capture
1. Sessions Service	1. Sessions Service	1. Sessions Service
2. Check Status Service	2. Check Status Service	2. Check Status Service
3. Order Service for 100.00	3. Order Service for 100.00	3. Order Service for 100.00
4. Sale Service for 100.00	4. Authorization Service for 70.00	4. Authorization Service for 70.00
	5. Capture Service for 70.00	5. Capture Service for 20.00
	6. Authorization Service for 30.00	6. Capture Service for 50.00
	7. Capture Service for 30.00	7. Authorization Service for 30.00
		8. Capture Service for 15.00
		9. Capture Service for 15.00

PayPal Express Order Tracking

Order tracking enables you to review transactions across your business and Cybersource reports by providing you with unique IDs - such as request IDs and reconciliation IDs that you can use for this purpose.

For all PayPal Express Checkout services, the request ID is returned in the response message in the **requestID** field. For more information on request IDs, see [Request IDs in Request Messages \(on page 36\)](#).

For more information about order tracking, see *Getting Started with Cybersource Advanced for the Simple Order API* ([PDF](#) | [HTML](#))

Reconciliation IDs

This table lists the fields for the reconciliationID values that are returned in individual service response messages and sent to PayPal as your invoice number. You can use these values to track transactions with Cybersource and PayPal.

Reconciliation ID Fields

Service	Field
Authorization	apAuthReply_reconciliationID
Authorization Reversal	apAuthReversalReply_reconciliationID
Cancel	apCancelReply_reconciliationID
Capture	apCaptureReply_reconciliationID
Check Status	apCheckStatusReply_reconciliationID
Order	apOrderReply_reconciliationID
Refund	apRefundReply_reconciliationID
Sale	apSaleReply_reconciliationID
Sessions	apSessionsReply_reconciliationID

Using PayPal Express Credit

 **Important:** PayPal Express Credit is supported only in the U.S.A.

The PayPal Express Credit button on your checkout page offers PayPal Express credit as a standalone option. Customers can defer payment by using the PayPal Express credit line included in their PayPal account. Do not display the PayPal Credit button on your shopping cart page if you require customers to log in before checking out.

! Important: The PayPal Express credit supports standard order flow (see [Using Standard Orders \(on page 12\)](#)) and custom order flows (see "Using Custom Orders," page 8 (on page 12)). Include the **apSessionsService_paymentOptionID** request field in the sessions request and set it to **credit**.

When your customers are ready to pay for your product, they follow these steps:

1. Click **PayPal Credit** on the checkout page.
2. Log in and enter a date of birth and Social Security number.
3. Agree to the PayPal credit terms and conditions.
4. Confirm the email address and credit line.
5. Confirm the credit amount. The customer is directed to the merchant web site.
6. Confirm order.

Adding the PayPal Express Credit Banner and Button to Your Page

1. Add the PayPal banner to advertise and explain the benefits of PayPal credit:
https://financing.paypal.com/docs/Web_Step_By_Step_Guide.pdf
2. Add the PayPal Credit button to your checkout page:
<https://www.paypal.com/us/webapps/mpp/logos-buttons>

Additional PayPal Express Checkout Services

Table 5: [Additional PayPal Express Checkout Services \(on page 15\)](#) below describes additional available services.

Additional PayPal Express Checkout Services

Service	Description
Authorization reversal service See Authorization Reversal Service (on page 27)	Reverses an authorization or an order setup: • When you perform multiple authorizations for an order and want to reverse one of them, you must specify which authorization to reverse by including the apAuthReversalService_authRequestID field.

Additional PayPal Express Checkout Services (continued)

Service	Description
Refund service See Refund Service (on page 30)	Refunds the capture amount to a customer. You can perform multiple partial refunds for each capture: • To refund an immediate full or partial sale, request the refund service and include the apRefundService_refundRequestID field. • To refund a delayed capture, you must specify which capture to credit by including the apRefundService_refundRequestID field.

Obtaining Transaction Information

The following table describes how to obtain information about your PayPal Express Checkout transactions.

Obtaining Transaction Information

Method	Description
PayPal Express Checkout services	Use the check status service (see Check Status Service (on page 20)).
Response messages	After you send a request message for a PayPal Express Checkout service, Cybersource responds with a response message that contains information about the status of your request, including errors and exceptions.
Reports	Use the Transaction Details Report and the Transaction Summary Report to manage your PayPal Express Checkout payments.
Transaction details	View the details of your PayPal Express Checkout transactions in the Business Center just as you can for other payment types. You can search for transactions by date, application type, customer name, and other transaction identifiers.

Dispute and Chargeback Services

This table describes PayPal Express Checkout services related to disputes and chargebacks.

Services for Disputes and Chargebacks

Service	Description
Chargeback dispute rights	When your customers choose a payment card as the funding source for the PayPal Express Checkout payment, they have chargeback dispute rights. If your customers dispute a charge, PayPal Express Checkout performs the initial chargeback processing and contacts you for documentation.
Buyer complaint process	PayPal Express Checkout offers a Buyer Complaint Process that applies to all purchases regardless of the funding source. When a customer files a complaint, PayPal investigates and contacts you for documentation. See http://www.paypal.com/cgi-bin/webscr?cmd=p/gen/buyer-complaint-outside .

Sessions Service

Requesting the Sessions Service

When your customer makes a purchase, you submit a sessions service request to begin the payment transaction. The sessions request can also be used to create a billing agreement.

The sessions service (**apSessionsService**):

- Starts the payment process or begins a billing agreement with PayPal, which sets up the customer order.
- To begin a billing agreement, you do not have to send any values, such as a dollar amount or a currency value. You also include additional fields, such as **ap_billingAgreementDescription** and **ap_billingAgreementIndicator**.
- Returns a Cybersource request ID (**requestID**) and a PayPal transaction ID (**apSessionsReply_processorTransactionID**).
- Returns a redirect URL (**apSessionsReply_merchantURL**) that includes a sessions token. Use the redirect URL to send the customer's browser to the PayPal web site.

Redirect URL: https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_expresscheckout&token=EC-2HF65459FN270362N

For additional information: <https://developer.paypal.com/docs/integration/direct/express-checkout/integrationjsv4/>.

For information on sessions service API fields, including billing agreement fields, see [API Fields \(on page 34\)](#).

Following the Sessions Request

- To check the status of the sessions request, use the check status service. See [Check Status Service \(on page 20\)](#).
- To continue placing a standard or custom order, use the order service. See [Order Service \(on page 24\)](#).
- To complete the process of creating a billing agreement with PayPal, use the billing agreement service. See [Billing Agreements Service \(on page 31\)](#).

For sessions service examples, see [Service Examples \(on page 73\)](#). For sessions service check status examples, see [Check Status Examples \(on page 92\)](#).

Check Status Service

Requesting the Check Status Service

You can check the status of an individual service request or of a billing agreement.

The check status service (**apCheckStatusService**):

- Requires the request ID (**requestID**) that was returned in the individual service response.
- Returns the status of the individual service request.
- If applicable, requires the billing agreement ID (**apReply_billingAgreementID**).
- If applicable, returns the customer's billing agreement details.

 **Important:** To get a customer's billing address details from PayPal, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.

Checking the Status of a Service Request

To check the status of a specific service request, you need to include the correct request ID. [Checking the Status of an Individual Service Request \(on page 21\)](#) describes the request IDs used to check the status of a service.

For more information on the request IDs used by each PayPal Express service, see [Request IDs in Request Messages \(on page 36\)](#).

 **Important:** If your individual service request included **item_#** fields, the **apCheckStatusService** service response returns **item_#** fields. To enable your account for **item_#** fields, contact Cybersource customer support.

For information on required and optional fields, see [API Fields \(on page 34\)](#).

For service examples, see [Service Examples \(on page 73\)](#). For check status examples, see [Check Status Examples \(on page 92\)](#).

Checking the Status of an Individual Service Request

1. Set the **apCheckStatusService_run** field to `true`.
2. Include the following required fields in the request:
 - **merchantID**
 - **merchantReferenceCode**
 - **apPaymentType**—set to `PPL`.
3. Set the **requestID** field based on the values described in [Table 8: Checking the Status of Individual Service Requests \(on page 21\)](#) below.

Checking the Status of Individual Service Requests

Service	Request ID Value
Authorization	To check the status of an authorization service request, set the apCheckStatusService_checkStatusRequestID field to the value returned in the requestID field of the authorization service response. See Authorization Service: Check Status (on page 93). The apCheckStatusReply_paymentStatus field returns one of the following values: • <code>AUTHORIZED</code> • <code>EXPIRED</code> • <code>FAILED</code> • <code>PENDING</code>
Billing Agreement	To check the status of a billing agreement service request, set the apCheckStatusService_sessionsRequestID field to the value returned in the requestID field of the sessions service response. Also in the billing agreement service check status request, set the ap_billingAgreementID field to the value returned in the apReply_billingAgreementID field in the billing agreement service response. See Check Status Request: Billing Agreement Service (on page 101). The apCheckStatusReply_paymentStatus field returns one of the following values: • <code>ACTIVE</code> • <code>CANCELLED</code>

Service	Request ID Value
	• FAILED
Capture	To check the status of a capture service request, set the apCheckStatusService_checkStatusRequestID field to the value returned in the requestID field of the capture service response. See Check Status Request: Capture Service (on page 94). The apCheckStatusReply_paymentStatus field returns one of the following values: • DISPUTED • FAILED • PENDING • SETTLED
Order	To check the status of an order service request, set the apCheckStatusService_checkStatusRequestID field to the value returned in the requestID field of the order service response. See Check Status Request: Order Service (on page 93). The apCheckStatusReply_paymentStatus field returns one of the following values: • CANCELLED • CREATED • FAILED
Refund	To check the status of a refund service request, set the apCheckStatusService_checkStatusRequestID field to the value returned in the requestID field of the refund service response. See Check Status Request: Refund Service (on page 95). The apCheckStatusReply_paymentStatus field returns one of the following values: • FAILED • PENDING • REFUNDED
Sale	To check the status of a sale service request, set the apCheckStatusService_checkStatusRequestID field to the value returned in the requestID field of the sale service response. See Check Status Request: Sale Service (on page 94).

Service	Request ID Value
	The apCheckStatusReply_paymentStatus field returns one of the following values: • DISPUTED • FAILED • PENDING • SETTLED
Sessions	To check the status of a sessions service request, set the apCheckStatusService_sessionsRequestID field to the value returned in the requestID field of the sessions service response. See Check Status Request: Sessions Service (on page 92). The apCheckStatusReply_paymentStatus field returns one of these status values: • CREATED • FAILED

Order Service

Requesting the Order Service

An order can be sent as a standard order (the full amount) or as a custom order (multiple partial amounts). For more information, see [PayPal Express Order Processing \(on page 12\)](#).

The order service (**apOrderService**):

- Requires the request ID (**requestID**) value returned in the sessions response.
- Requires the payer ID (**ap_payerID**) value to complete the payment.
- Creates an order in anticipation of one or more authorizations.
- Returns a transaction ID

Following the Order Request

- To continue with a standard order, use the sale service. See [Sale Service \(on page 29\)](#).
- To continue with a custom order, use the authorization service. See [Authorization Service \(on page 26\)](#).
- To void a standard or custom order, use the cancel service. See [Cancel Service \(on page 25\)](#).

 **Important:** Once a session service request has been submitted, the subtotal amount cannot be updated. If there is a change in any of the amount fields, all of the amount fields must be sent in a new request. They must equal the grand total amount.

For required and optional fields, see [API Fields \(on page 34\)](#).

For order service examples, see [Service Examples \(on page 73\)](#). For order service check status examples, see [Check Status Examples \(on page 92\)](#).

Cancel Service

Requesting the Cancel Service

To void a standard or custom order, use the cancel service.

 **Important:** You cannot cancel an order once it has been authorized or captured.

The cancel service (**apCancelService**):

- Voids or cancels the order.
- Requires the request ID (**requestID**) value that was returned in the order response. Include the value in the **apCancelService_orderRequestID** field.
- When processing a billing agreement transaction, include the **ap_billingAgreementID** field. See [Billing Agreements Service \(on page 31\)](#).

For required and optional fields, see [API Fields \(on page 34\)](#).

For cancel service examples, see [Service Examples \(on page 73\)](#) and [Check Status Examples \(on page 92\)](#).

Authorization Service

Requesting the Authorization Service

The full amount of your order can be authorized for a standard order, or you can send multiple partial authorizations for a custom order.

The authorization service (**apAuthService**):

- Requires the request ID (**requestID**) value that was returned in the order response.
- Obtains the authorization.
- Enables you to perform multiple partial authorizations.

The authorization service commits funds for three business days. For three days you can try to capture the funds. If the funds are available, the capture is successful.

 **Important:** If three days is not enough to successfully capture your custom orders, contact PayPal to have the honor interval increased. For more information on custom orders, see [Using Custom Orders \(on page 12\)](#).

Following the Authorization Request

- To capture the authorized amount, use the capture service. See [Capture Service \(on page 28\)](#).
- To reverse the authorization, use the authorization reversal service. See [Authorization Reversal Service \(on page 27\)](#).

For required and optional fields, see [API Fields \(on page 34\)](#).

For authorization service examples, see [Service Examples \(on page 73\)](#). For authorization service check status examples, see [Check Status Examples \(on page 92\)](#).

Authorization Reversal Service

Requesting the Authorization Reversal Service

Reversing an authorization removes the hold placed on your customer's funds. Both full and partial authorizations can be reversed.

The authorization reversal service (**apAuthReversalService**):

- Requires the request ID (**requestID**) value that was returned in the authorization service response.

 **Important:** Reversing an authorization on funds that have been captured does not release the captured amount. To return a customer's funds once they have been captured, use the refund service.

For required and optional fields, see [API Fields \(on page 34\)](#).

For authorization reversal service examples, see [Service Examples \(on page 73\)](#). For authorization reversal check status service examples, see [Check Status Examples \(on page 92\)](#).

Capture Service

Requesting the Capture Service

The capture service follows an authorization request and transfers funds to your account to complete the payment transaction.

The capture service (**apCaptureService**):

- Requires the request ID (**requestID**) value that was returned in the authorization response.
- Enables you to capture the entire authorized amount.
- Enables you to perform multiple partial captures (called *split shipments*) of up to 115% of the authorized amount.

Following the Capture Request

To refund the customer's funds, use the refund service. See [Refund Service \(on page 30\)](#).

For required and optional fields, see [API Fields \(on page 34\)](#).

For capture service examples, see [Service Examples \(on page 73\)](#). For capture service check status examples, see [Check Status Examples \(on page 92\)](#).

Sale Service

Requesting the Sale Service

The sale service is an authorization and capture in one request. The sale service transfers funds to your account to complete the payment transaction.

The sale service (**apSaleService**):

- Requires the request ID value returned in the order response. Include this value in the **apSaleService_orderRequestID** field.
- Requires the billing agreement ID value, if you are processing a billing agreement. Include this value in the **ap_billingAgreementID** field. See [Billing Agreements Service \(on page 31\)](#).
- Returns a processor transaction ID (**apSaleReply_processorTransactionID**).

Following the Sale Request

To refund the customer's funds, use the refund service. See [Refund Service \(on page 30\)](#).

For required and optional fields, see [API Fields \(on page 34\)](#).

For sale service examples, see [Service Examples \(on page 73\)](#). For sale service check status examples, see [Check Status Examples \(on page 92\)](#).

Refund Service

Requesting the Refund Service

To return funds to the customer after the payment transaction is completed, use the refund service.

The refund service (**apRefundService**):

- Requires the request ID (**requestID**) value that was returned in the capture response or the sale response.
- Enables you to perform a full refund or multiple partial refunds for an order.

For required and optional fields, see [API Fields \(on page 34\)](#).

For refund service examples, see [Service Examples \(on page 73\)](#). For refund service check status examples, see [Check Status Examples \(on page 92\)](#).

Billing Agreements Service

Requesting the Billing Agreements Service

A billing agreement is set up between PayPal and your customer. When you collect the details of a customer's billing agreement, you are able to bill that customer without requiring an authorization for each payment. You can bill the customer at the same time you process their PayPal Express checkout order, which simplifies your business processes.

A billing agreement is also called a reference transaction. A billing agreement contains the customer's billing information, such as first and last name, street address, city, country, postal code, and more.

 **Important:** To get a customer's billing address details from PayPal, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.

Setting Up a Billing Agreement

Your merchant account must be configured to process billing agreements.

- For a Production account—contact your PayPal account manager to configure your production account or contact PayPal Business Support.
- For a Sandbox account—If your PayPal sandbox account was created before December 15, 2015, contact your PayPal account manager or [PayPal merchant technical support](#). If your PayPal sandbox account was created on or after December 15, 2015, it is already configured to process billing agreements.

Creating a Billing Agreement

1. Request the sessions service to receive the redirection URL (**apSessionsReply_merchantURL**) that directs the customer to PayPal to confirm their billing agreement.

To create a billing agreement using the sessions service, you can do it in two ways:

- As a sale transaction. In this case, you request the sessions service using the customer's sale information and the the billing agreement information.

- As a billing agreement transaction. In this case, you do not need to send a dollar amount or any currency value. These fields can be zero.

For more information, see [Sessions Service \(on page 18\)](#).

2. Request the billing agreement service and include the request ID (**requestID**) value returned in the sessions service response.
3. Request the check status service to receive the customer's billing agreement details from PayPal. The check status request requires these two values:
 - The request ID (**requestID**) value returned in the sessions service response.
 - The billing agreement ID (**apReply_billingAgreementID**) value returned in the billing agreement service response.

For more information, see [Check Status Service \(on page 20\)](#).

4. Request the sale service to process the customer's billing agreement. You can do this in two ways:
 - As a sale transaction, with a reference transaction:
 - Include the billing agreement information created in [Step 2 \(on page 32\)](#).
 - Include the billing agreement ID (**apReply_billingAgreementID**) value returned in the billing agreement service response.
 - When you submitted the order service request, it returned the order ID (**apReply_orderID**) value. Include this value in your sale service request.
 - As a reference transaction only:
 - Include the billing agreement information created in [Step 2 \(on page 32\)](#).
 - Include the billing agreement ID (**apReply_billingAgreementID**) value returned in the billing agreement service response.

For more information, see [Sale Service \(on page 29\)](#).

For required and optional fields, see [API Fields \(on page 34\)](#).

Billing Agreement Service Examples

The following examples help illustrate how billing agreements are used with these services:

- Sessions service—you can create a billing agreement using the sessions service and then check its status using the check status service. In this flow, you can begin a sale transaction and create a billing agreement in the same request. For sessions service examples, see [Sessions Service \(on page 18\)](#). For sessions service check status examples, see [Check Status Examples \(on page 92\)](#).
- Billing agreement service—you can create a billing agreement using the billing agreement service and then check its status using the check status service. In this flow, you create a billing agreement without any sale transaction. For billing agreement service examples, see [Billing Agreement Service \(on page 90\)](#). For billing agreement service check status examples, see [Check Status Examples \(on page 92\)](#).
- Sale service—you can include billing agreement fields in the sale service to create a reference transaction. For sale service reference transaction examples, see [Sale Service \(on page 29\)](#). For sale service check status examples, see [Check Status Examples \(on page 92\)](#).

API Fields

Formatting Restrictions

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

The values of the **item_#_** fields must not contain carets (^) or colons (:) because these characters are reserved for use by the 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. All leading and trailing spaces are removed.

For Moneris, values for request-level and item-level fields must not contain these special characters: ampersands (&), single quotes ('), double quotes ("), less-than signs (<), and greater-than signs (>).

Data Type Definitions

Numbered Elements

The Cybersource XML schema includes several numbered elements. You can include these complex elements more than once in a request. For example, when a customer order includes more than one item, you must include multiple `<item>` elements in your request. Each item is numbered, starting with 0. The XML schema uses an `id` attribute in the item's opening tag to indicate the number. For example:

```
<item id="0">
```

As a name-value pair field name, this tag is called `item_0`. In this portion of the field name, the underscore before the number does not indicate hierarchy in the XML schema. Each item field is generically referred to as `item_#_<element name>` in the documentation.

Below is an example of the numbered `<item>` element and the corresponding name-value pair field names. If you are using the Simple Object Access Protocol (SOAP), the client contains a corresponding item class.

XML Schema Element Names	Corresponding Name-Value Pair Field Names
<pre><item id="0"> <unitPrice> <quantity> </item></pre>	<pre>item_0_unitPrice item_0_quantity</pre>
<pre><item id="1"> <unitPrice> <quantity> </item></pre>	<pre>item_1_unitPrice item_1_quantity</pre>

 **Important:** When a request in XML format includes an `<item>` element, the element must include an `id` attribute. For example: `<item id="0">`.

Working with Multibyte Character Strings

Some languages—such as Korean, Chinese, and Russian—require more than one ASCII byte per character. Therefore, a string of multibyte characters can exceed the allotted string length of some Cybersource API fields.

In some languages, a short name that has only three or four characters might be equivalent to 10 or 15 bytes in UTF-8. Therefore, it is important to consider the overall length of a multibyte string to be sure that it fits into the allotted ASCII string length.

For best success when working with languages, make sure that the total length of a multibyte string does not exceed the ASCII string length for a specific API field.

In some languages, a short name that has only three or four characters might be equivalent to 10 or 15 bytes in UTF-8. Therefore, it is important to consider the overall length of a multibyte string to be sure that it fits into the allotted ASCII string length.

For best success when working with languages, make sure that the total length of a multibyte string does not exceed the ASCII string length for a specific API field.

Request IDs in Request Messages

The following table lists the request ID field names used in request messages. For the request ID field names used to check the status of an individual service, see [Checking the Status of an Individual Service Request \(on page 21\)](#).

Request IDs in Request Messages

Service Request	Request Field Names
Authorization	• apAuthService_orderRequestID—set to the value of the requestID field returned in the order service response.
Authorization Reversal	• apAuthReversalService_authRequestID—set to the value of the requestID field returned in the authorization service response.
Billing Agreement (reference transaction)	• apBillingAgreementService_sessionsServiceID—set to the value of the requestID field returned in the sessions service response. • apCheckStatusService_checkStatusRequestID—set to the value of the requestID field returned in the billing agreement service response.
Cancel	• apCancelService_orderRequestID—set to the value of the requestID field returned in the order service response. • ap_billingAgreementID—set to the value of the requestID returned billing agreement service response.
Capture	• apCaptureService_authRequestID—set to the value of the requestID field returned in the authorization service response.
Check Status (get payment details)	• apCheckStatusService_sessionsRequestID—set to the value of the requestID returned in the sessions service response. • apCheckStatusService_checkStatusRequestID—set to the value of the requestID field returned in the: ◦ Authorization service response ◦ Billing agreement service response ◦ Capture service response

Request IDs in Request Messages (continued)

Service Request	Request Field Names
	◦ Order service response ◦ Refund service response ◦ Sale service response
Order	• apOrderService_sessionsRequestID—set to the value of the requestID field returned in the sessions service response. • apCheckStatusService_checkStatusRequestID—set to the value of the requestID field returned in the order service response.
Refund	• apRefundService_refundRequestID—set to the value of the requestID field returned in the capture service response. • apRefundService_refundRequestID—set to the value of the requestID field returned in the sale service response.
Sale	• apSaleService_orderRequestID—set to the value of the requestID returned in the order service response. or • ap_billingAgreement—include this field if processing a billing agreement.

Request IDs in API Response Messages

For all PayPal Express Checkout services, the request ID is returned in the response message in the **requestID** field.

Request Fields

Request Fields

Field	Description	Required (R) / Optional (O)	Data Type & Length
apAuthReversalService_authRequestID	Set to the value of the requestID field returned in the authorization service response.	Auth Reversal (R)	String (26)
apAuthService_orderRequestID	Set to the value of the requestID field returned in the order service response.	Authorization (R)	String (26)
apAuthService_run	The authorization service request name.	Authorization (R)	String (26)
ap_billingAgreementDescription	Description of the billing agreement.	Sessions (O for billing agreement)	String (127)
ap_billingAgreementID	The identifier for the billing agreement. Set to the value returned in the apReply_billingAgreementID field.	Cancel (O) Check Status (R when checking the status of a billing agreement) Sale (R for a reference transaction)	String (50)
ap_billingAgreementIndicator	Indicates whether the transaction is a billing agreement. Possible values: - FALSE (default) - TRUE	Billing Agreement (R) Sessions (O for a billing agreement)	String (5)
apBillingAgreementService_run	The billing agreement service request name.	Billing Agreement (O)	String (127)
apBillingAgreementService_sessionsRequestID	Set to the value of the requestID field returned in the sessions service response.	Billing Agreement (R)	String (26)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
apCancelService_orderRequestID	Set to the value of the requestID field returned in the order service response.	Cancel (R)	String (26)
apCancelService_run	The cancel service request name.	Cancel (R)	String (26)
apCaptureService_isFinal	Indicates whether to release the authorization hold on the remaining funds. Possible values: • FALSE (default) • TRUE	Capture (O)	String (5)
apCaptureService_run	The capture service request name.	Capture (O)	String (26)
apCheckStatusService_run	The check status service request name.	Check Status (R)	String (26)
apCheckStatusService_checkStatusRequestID	Request ID of the transaction for which you want to check the status. This value is returned in the service response you want to check.	Check Status (R)	String (26)
apCheckStatusService_sessionsRequestID	Set to the value of the requestID field returned in the sessions service response.	Check Status (R)	String (26)
ap_fundingSource	Payment method for the unit purchase. Possible values: • INSTANT • UNRESTRICTED (default)—this value is available only when configured by PayPal for the merchant.	Refund (O) Sale (O)	String (30)
ap_payerID	The ID of the customer.	Order (R)	String(30)
apPaymentType	Identifier for the payment type. Value: PPL	Authorization (R)	String (3)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
		Auth Reversal (R) Billing Agreement (R) Cancel (R) Capture (R) Check Status (R) Order (R) Refund (R) Sale (R) Sessions (R)	
apOrderService_run	The order service request name.	Order (R)	String (26)
apOrderService_sessionsRequestID	Set to the value of the requestID field returned in the sessions service response.	Order (R)	String (26)
apRefundService_reason	The reason for the refund.	Refund (O)	String (30)
apRefundService_refundRequestID	Set to the value of the requestID field returned in the sale or capture service response.	Refund (R)	String (26)
apRefundService_run	The refund service request name.	Refund (R)	String (26)
apSaleService_orderRequestID	Set to the value of the requestID field returned in the order service response.	Sale (R)	String (26)
apSaleService_run	The sale service request name.	Sale (R)	String (26)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
apSessionsService_cancelURL	URL to which the customer is directed after canceling the payment. NOTE: Cybersource recommends that you work with customer support to get this field enabled in your merchant configuration.	Sessions (O only if already configured in merchant configuration ; otherwise R)	String (255)
apSessionsService_paymentOptionID	Identifier for a PayPal credit transaction. Value: Credit	Sessions (R for PayPal credit transaction only)	String (255)
apSessionsService_run	The sessions service request name.	Sessions (R)	String (26)
apSessionsService_successURL	URL to which the customer is directed after completing the payment. Cybersource recommends that you work with customer support to get this field enabled in your merchant configuration.	Sessions (O only if already configured in merchant configuration ; otherwise R)	String (255)
clientMetadataID	Verifies that the payment is originating from a valid, user-approved application and device. Sending this field helps reduce fraud and declined transactions.  Note: The length is set for a hexadecimal representation of the GUID/UUID. This field accepts a 36-character string (with hyphens) or a 32-character string (without hyphens). 123e4567-e89b-12d3-a456-426655440000	Sale (R for reference transactions; otherwise O)	String (36)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
	123e4567e89b12d3a456426655440000		
invoiceHeader_merchantDescriptor	Merchant description on the customer's statement. When you include more than one consecutive space, extra spaces are removed.	Sale (O) Sessions (O)	String (140)
item_#_productCode	Type of product. This value is used to determine the product category: electronic, handling, physical, service, or shipping. The default value is default . For the authorization service, when you set this field to a value other than default or any of the values related to shipping and handling, the item_#_quantity, item_#_productName, and item_#_productSKU fields are required.	Sale (O; only for reference transactions) Sessions (O when creating an order; otherwise not used)	String (255)
item_#_productDescription	Description of the product.  Note: Do not include this field when creating a billing agreement.	Sale (O; only for reference transactions) Sessions (O when creating an order; otherwise not used)	String (255)
item_#_productName	Name of the product. For the authorization service and capture service, this field is required when item_#_productCode is not default or one of the values related to shipping and handling.	Sale (O; only for reference transactions) Sessions (O when creating an order; otherwise not used)	String (255)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
	 Note: Do not include this field when creating a billing agreement.		
item_#_productSKU	Identification code for the product. For the authorization service and capture service, this field is required when item_#_productCode is not <code>default</code> or one of the values related to shipping and handling.  Note: Do not include this field when creating a billing agreement.	Sale (O; only for reference transactions) Sessions (O when creating an order; otherwise not used)	String (255)
item_#_quantity	The default is <code>1</code>. For the authorization service and capture service, this field is required when item_#_productCode is not <code>default</code> or one of the values related to shipping and handling.  Note: Do not include this field when creating a billing agreement.	Sale (O; only for reference transactions) Sessions (O when creating an order; otherwise not used)	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: You include the following items in your request:	Sale (O; only for reference transactions)	String (15)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
	<pre> 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 </pre> The total amount authorized is 32.40, not 30.00 with 2.40 of tax included.		
item_#_unitPrice	Per-item price of the product. 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.  Note: Do not include this field when creating a billing agreement.	Sale (O; only for reference transactions) Sessions (O when creating an order; otherwise not used)	String (15)
item_#_unitTaxAmount	Per-item tax amount of the product. The amount value must be a non-negative number containing 2 decimal places and limited to 7 digits before the decimal point.  Note: Do not include this field when creating a billing agreement.	Sale (O; only for reference transactions) Sessions (O when creating an order; otherwise not used)	String (15)
merchantID	Your Cybersource merchant ID. Use the same merchant ID for evaluation, testing, and production.	Required for all services	String (30)
merchantReferenceCode	Merchant-generated transaction number or tracking number. Cybersource recommends that you send a unique value for each transaction so that you can perform	Required for all services	String (50)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
	meaningful searches for the transaction. For information about tracking orders and transactions, see Getting Started with Cybersource Advanced for the Simple Order API .		
note_toPayee	Note to the recipient of the funds in this transaction.	Sale (O)	String (255)
note_toPayer	Free-form text field.	Sale (O)	String (165)
purchaseTotals_currency	Currency for the transaction. Use the three-character ISO Standard Currency Codes.  Important: Decimal places are not supported for the Hungarian Forint (HUF) currency code.  Note: Do not include this field when creating a billing agreement.	Authorization (R) Capture (R) Order (R) Refund (R) Sale (R) Sessions (R when creating an order)	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. The amount value must be a non-negative number containing 2 decimal places and limited to 7 digits before the decimal point.  Note: Do not include this field when creating a billing agreement.	Authorization (R) Capture (R) Order (R) Refund (R) Sale (R) Sessions (R when creating an order)	String (10)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
purchaseTotals_handlingAmount	Aggregate handling charges for the transaction. If this amount has changed since the initial sessions request, you must include the new value in the order request. You must also include all additional amount fields that apply to the order and ensure the total amount equals the purchaseTotals_grandTotalAmount value.  Note: The amount value must be a non-negative number containing 2 decimal places and limited to 7 digits before the decimal point.	Order (O) Sale (O; only for a reference transaction) Sessions (R when creating an order)	String (15)
purchaseTotals_shippingAmount	Aggregate shipping charges for the transaction. If this amount has changed since the initial sessions request, you must include the new value in the order request. You must also include all additional amount fields that apply to the order and ensure the total amount equals the purchaseTotals_grandTotalAmount value.  Note: The amount value must be a non-negative number containing 2 decimal places and limited to 7 digits before the decimal point.	Order (O) Sale (O; only for a reference transaction) Sessions (R when creating an order)	String (15)
purchaseTotals_shippingDiscountAmount	Shipping discount amount for the transaction.	Order (O) Sale (O)	String (15)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
	If this amount has changed since the initial sessions request, you must include the new value in the order request. You must also include all additional amount fields that apply to the order and ensure the total amount equals the purchaseTotals_grandTotalAmount value.  Note: The amount value must be a negative number containing 2 decimal places and limited to 7 digits before the decimal point.	Sessions (O)	
purchaseTotals_subtotalAmount	Subtotal amount of all items. This amount (which is the value of all items in the cart, not including the additional amounts such as tax, shipping, etc.) cannot change after a sessions request. When there is a change to any of the additional amounts, this field should be resent in the order request. When the subtotal amount changes, you must initiate a new transaction starting with a sessions request.  Note: The amount value must be a non-negative number containing 2 decimal places and limited to 7 digits before the decimal point. This value can not be changed after a sessions request.	Order (O) Sale (O; only for a reference transaction) Sessions (R when creating an order)	String (15)
purchaseTotals_taxAmount	Total tax amount. When the purchaseTotals_taxAmount and ap_subtotalAmount fields are	Order (O)	String (10)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
	included in the request, do not include the tax amount as part of the subtotal amount calculation.  Note: The amount value must be a non-negative number containing 2 decimal places and limited to 7 digits before the decimal point.	Sessions (R when creating an order)	
shipTo_city	City of the shipping address.  Note: This field is optional and valid only for a billing agreement (reference transaction).	Sale (O; only for a reference transaction) Sessions (O)	String (40)
shipTo_country	Country of the shipping address. Use the two-character ISO Standard Country Codes.  Note: This field is optional and valid only for a billing agreement (reference transaction).	Sale (O; only for a reference transaction) Sessions (O)	String (2)
shipTo_firstName	First name of the recipient.  Note: This field is optional and valid only for a billing agreement (reference transaction). In-Store Collection When processing an in-store collection transaction in which the customer orders online and picks up the order in the store, avoid PayPal fraud declines by including s2s as the first 3	Sale (O; only for a reference transaction) Sessions (O)	String (60)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
	characters. A blank space must also be included between S2S and the rest of the first name information. S2S John		
shipTo_immutable	Indicates whether customers are permitted to edit the shipping address in their PayPal account. Possible values: FALSE (default): Customer can edit the shipping address. TRUE: Customer cannot edit the shipping address. When TRUE, the merchant-provided shipping address is used and the buyer cannot change the shipping address on the PayPal pages. If the merchant does not pass a shipping address, the buyer can choose the shipping address on PayPal pages.	Sale (O; only for a reference transaction) Sessions (O)	String (5)
shipTo_lastName	Last name of the recipient.  Note: This field is optional and valid only for a billing agreement (reference transaction).	Sale (O; only for a reference transaction) Sessions (O)	String (60)
shipTo_notApplicable	Indicates whether the shipping address is displayed to the customer in their PayPal account. Possible values: FALSE (default): Shipping address is displayed. TRUE: Shipping address is not displayed..	Sale (O; only for a reference transaction) Sessions (O)	String (5)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
	For example, for digital downloads and services in which a shipping address is not required, set the value to true .		
shipTo_phoneNumber	Phone number for the shipping address.  Note: This field is optional and valid only for a billing agreement (reference transaction).	Sale (O; only for a reference transaction) Sessions (O)	String (20)
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] 12345-6789 When the shipping country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space][numeric][alpha][numeric] A1B 2C3  Note: This field is optional and valid only for a billing agreement (reference transaction).	Sale (O; only for a reference transaction) Sessions (O)	String (20)
shipTo_shippingMethod	Shipping method for the product. Possible values: • SAMEDAY: Courier or same -day service.	Sale (O; only for a reference transaction)	String (10)

Request Fields (continued)

Field	Description	Required (R) / Optional (O)	Data Type & Length
	• ONEDAY: Next-day or overnight service. • TWODAY: Two-day service. • THREEDAY: Three-day service. • LOWCOST: lowest-cost service. • PICKUP: Store pick-up. • OTHER: Other shipping method. • NONE: No shipping method because the product is a service or subscription.	Sessions (O)	
shipTo_state	State or province of the shipping address. Use the State, Province, and Territory Codes for the United States and Canada.  Note: This field is optional and valid only for a billing agreement (reference transaction).	Sale (O; only for a reference transaction) Sessions (O)	String (40)
shipTo_street1	First line of the shipping address.  Note: This field is optional and valid only for a billing agreement (reference transaction).	Sale (O; only for a reference transaction) Sessions (O)	String (100)
shipTo_street2	Second line of the shipping address.  Note: This field is optional and valid only for a billing agreement (reference transaction).	Sale (O; only for a reference transaction) Sessions (O)	String (100)

Response Fields

Response Fields

Field	Description	Returned By	Data Type & Length
apAuthReply_amount	The authorized amount.	Authorization	String (15)
apAuthReply_dateTime	Time of authorization. Format: YY YY-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Authorization	String (18)
apAuthReply_paymentStatus	Authorization status from the processor. Possible values: • AUTHORIZED • EXPIRED • FAILED • PENDING	Authorization	String (15)
apAuthReply_processorResponse	The processor code that describes why the transaction state is pending or reversed.  Note: The returned value can also include the processor response values. See Reason Codes (on page 103).	Authorization	String (60)
apAuthReply_processorTransactionID	Identifier of the order transaction. Read-only.	Authorization	String (50)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
apAuthReply_reasonCode	Numeric value corresponding to the result of the authorization service request.	Authorization	Integer (5)
apAuthReply_reconciliationID	Reference number that you use to reconcile your Cybersource reports with your business reports.	Authorization	String (60)
apAuthReply_responseCode	Response code from the processor.	Authorization	Integer (5)
apAuthReversalReply_amount	The authorized amount that was reversed.	Auth Reversal	String (15)
apAuthReversalReply_dateTime	Date and time of authorization reversal. Format: yyyy-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Auth Reversal	String (18)
apAuthReversalReply_paymentStatus	The payment status returned from the payment processor. Possible values: • AUTH_REVERSED • FAILED	Auth Reversal	String (15)
apAuthReversalReply_processorResponse	The processor code that describes why the transaction state is pending or reversed.  Note: The returned value can also include the processor response values. See Reason Codes (on page 103) .	Auth Reversal	String (60)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
apAuthReversalReply_processorTransactionID	Identifier of the order transaction. Read-only.	Auth Reversal	String (50)
apAuthReversalReply_reasonCode	Numeric value corresponding to the result of the auth reversal service request.	Auth Reversal	Integer (5)
apAuthReversalReply_reconciliationID	Reference number that you use to reconcile your Cybersource reports with your business reports.	Auth Reversal	String (60)
apAuthReversalReply_responseCode	Response code from the processor.	Auth Reversal	Integer (5)
apBillingAgreementReply_amount	Total amount charged from the payer to the payee. Maximum of 10 characters with support for 2 decimal places.	Billing Agreement	String(15)
apBillingAgreementReply_dateTime	Time of billing agreement. Format: yyyy-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Billing Agreement	String (18)
apBillingAgreementReply_processorResponse	Response code indicating that creating the agreement failed.  Note: The returned value can also include the processor response values. See Reason Codes (on page 103) .	Billing Agreement	String(60)
apBillingAgreementReply_reasonCode	Numeric value corresponding to the result of the billing agreement service request.	Billing Agreement	Integer (5)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
apBillingAgreementReply_status	State of the order transaction. Possible values: • ACTIVE • FAILED • INACTIVE	Billing Agreement	String(60)
apCancelReply_dateTime	Time of cancel. Format: yyyy-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Cancel	String (18)
apCancelReply_processorResponse	The reason when the transaction status is Pending or Reversed. Possible values: • MULTI_CURRENCY • ORDER • OTHER • PAYER_SHIPPING_UNCONFIRMED • REGULATORY_REVIEW • RISK_REVIEW • VERIFICATION_REQUIRED	Cancel	String (60)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	 Note: The returned value can also include the processor response values. See Reason Codes (on page 103) .		
apCancelReply_processorTransactionID	Identifier of the order transaction. Read-only.	Cancel	String (50)
apCancelReply_reasonCode	Numeric value corresponding to the result of the cancel service request. See Reason Codes (on page 103) .	Cancel	Integer (5)
apCancelReply_reconciliationID	Reference number that you use to reconcile your Cybersource reports with your reports.	Cancel	String (60)
apCancelReply_status	CYBS mapped values. Possible values: • CANCELED • FAILED	Cancel	String (60)
apCaptureReply_reasonCode	Numeric value corresponding to the result of the capture service request.	Capture	Integer (5)
apCaptureReply_dateTime	Time of capture. Format: yyyy-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Capture	String (18)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
apCaptureReply_paymentStatus	Description of the payment status. Possible values: • FAILED • PENDING • SETTLED	Capture	String (15)
apCaptureReply_processorResponse	The processor code that describes why the transaction state is pending or reversed. Possible values when the payment status is pending: • INTERNATIONAL_WITHDRAWAL • PAYMENT_REVIEW • RECEIVING_PREFERENCE_MANDATES_MANUAL_ACTION • REGULATORY_REVIEW • TRANSACTION_APPROVED_AWAITING_FUNDING • UNCONFIRMED_SHIPPING_ADDRESS • VERIFICATION_REQUIRED  Note: The returned value can also include the processor response values. See Reason Codes (on page 103).	Capture	String (60)
apCaptureReply_processorTransactionFee	Amount up to N digit after the decimal separator as defined in ISO 4217 for the appropriate currency code.	Capture	String (15)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
apCaptureReply_processorTransactionID	Identifier of the order transaction. Read-only.	Capture	String (50)
apCaptureReply_reconciliationID	Reference number that you use to reconcile your Cybersource reports with your business reports.	Capture	String (60)
apCheckStatusReply_reasonCode	Numeric value corresponding to the result of the check status service request.	Check Status	Integer (5)
apCheckStatusReply_dateTime	Time of check status. Format: yyyy-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Check Status	String (18)
apCheckStatusReply_paymentStatus	Description of the payment status. Possible values: • CREATED • FAILED	Check Status	String (15)
apCheckStatusReply_processorResponse	The value returned by the processor indicating the reason the request failed. See Reason Codes (on page 103) .	Check Status	String (60)
apCheckStatusReply_processorTransactionID	Identifier of the order transaction. Read only.	Check Status	String (50)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
apCheckStatusReply_reconciliationID	Reference number that you use to reconcile your Cybersource reports with your business reports.	Check Status	String (60)
apOrderReply_amount	Total amount charged from the customer to the merchant. In the case of a refund, this is the amount refunded to the customer from the merchant. Ten-character maximum with 0 to 2 decimal places.	Order	String (15)
apOrderReply_dateTime	Time of order. Format: yyyy-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Order	String (18)
apOrderReply_processorResponse	The reason for the payment failure. Possible values: • CANNOT_PAY_THIS_PAYEE • INVALID_PAYMENT_METHOD • MULTI_CURRENCY • ORDER • OTHER • PAYEE_FILTER_RESTRICTIONS • PAYER_CANNOT_PAY • PAYER_SHIPPING_UNCONFIRMED • REDIRECT_REQUIRED	Order	String (60)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	• REGULATORY_REVIEW • RISK_REVIEW • UNABLE_TO_COMPLETE_TRANSACTION • VERIFICATION_REQUIRED  Note: The returned value can also include the processor response values. See Reason Codes (on page 103).		
apOrderReply_reasonCode	Numeric value corresponding to the result of the order service request. See Reason Codes (on page 103) .	Order	Integer (5)
apOrderReply_reconciliationID	Invoice number to track the payment.	Order	String (127)
apOrderReply_status	State of the order transaction. Possible values: • CANCELLED • CREATED • FAILED	Order	String (60)
apRefundReply_dateTime	Time of refund. Format: yyyy-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Refund	String (18)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
apRefundReply_paymentStatus	Refund status from the processor. Possible values: • FAILED • PENDING • REFUNDED	Refund	String (15)
apRefundReply_processorResponse	The processor code that describes why the transaction state is pending or reversed.  Note: The returned value can also include the processor response values. See Reason Codes (on page 103).	Refund	String (60)
apRefundReply_processorTransactionID	Identifier of the order transaction. Read-only.	Refund	String (50)
apRefundReply_reasonCode	Numeric value corresponding to the result of the refund service request. See Reason Codes (on page 103).	Refund	Integer (5)
apRefundReply_reconciliationID	Reference number that you use to reconcile your Cybersource reports with your business reports.	Refund	String (60)
apReply_billingAgreementID	The Billing Agreement ID returned by processor (PayPal).	Billing Agreement	String(50)
apReply_fundingSource	Payment mode for the transaction. Possible values:	Authorization Check Status Order	String (30)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	• DELAYED_TRANSFER • ECHECK • INSTANT_TRANSFER • MANUAL_BANK_TRANSFER	Sale Sessions	
apReply_orderID	Identifier for the purchase.	Order	String (20)
apReply_payerID	The ID of the customer, passed in the return_url field by PayPal after customer approval.	Billing Agreement Check Status Order	String(30)
apReply_processorFraudDecision	Type of filter. Possible values: • ACCEPT • DENY • PENDING • REPORT	Authorization Order Sale	String (60)
apReply_processorFraudDecisionReason	Filter Identifier. Possible values: • AVS_NO_MATCH • AVS_PARTIAL_MATCH • AVS_UNAVAILABLE_OR_UNSUPPORTED • BILLING_OR_SHIPPING_ADDRESS_MISMATCH • CARD_SECURITY_CODE_MISMATCH • COUNTRY_MONITOR • IP_ADDRESS_VELOCITY • LARGE_ORDER_NUMBER	Authorization Order Sale	String (60)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	• <code>MAXIMUM_TRANSACTION_AMOUNT</code> • <code>PAYPAL_FRAUD_MODEL</code> • <code>RISKY_BANK_IDENTIFICATION_NUMBER_CHECK</code> • <code>RISKY_EMAIL_ADDRESS_DOMAIN_CHECK</code> • <code>RISKY_IP_ADDRESS_RANGE</code> • <code>RISKY_ZIP_CODE</code> • <code>SUSPECTED_FREIGHT_FORWARDER_CHECK</code> • <code>TOTAL_PURCHASE_PRICE_MINIMUM</code> • <code>UNCONFIRMED_ADDRESS</code>		
<code>apReply_sellerProtection_eligibility</code>	The level of seller protection in force for the transaction. Possible values: • <code>ELIGIBLE</code> • <code>INELIGIBLE</code> • <code>PARTIALLY_ELIGIBLE</code>	Authorization Check Status Order Sale	String (60)
<code>apReply_sellerProtection_type</code>	The kind of seller protection in force for the transaction. This field is returned only when the protection_eligibility property is set to <code>ELIGIBLE</code> or <code>PARTIALLY_ELIGIBLE</code>. Possible values: • <code>ITEM_NOT_RECEIVED_ELIGIBLE</code>—Sellers are protected against claims for items not received.	Authorization Check Status Order Sale	String (60)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	• UNAUTHORIZED_PAYMENT_ELIGIBLE—Sellers are protected against claims for unauthorized payments. One or both values can be returned.		
apSaleReply_amount	Total amount charged from the payer to the payee. Maximum of 10 characters with support for 2 decimal places.	Sale	String (15)
apSaleReply_dateTime	Time of sale. Format: yyyy-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Sale	String (18)
apSaleReply_paymentStatus	Description of the payment status. Possible values: • FAILED • PENDING • SETTLED	Sale	String (15)
apSaleReply_processorResponse	The processor code that describes why the transaction state is pending or reversed. Possible values: • BUYER_COMPLAINT • CHARGEBACK • ECHECK	Sale	String (60)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	• GUARANTEE • INTERNATIONAL_WITHDRAWAL • PAYMENT_REVIEW • RECEIVING_PREFERENCE_MANDATES_MANUAL_ACTION • REGULATORY_REVIEW • TRANSACTION_APPROVED_AWAITING_FUNDING • UNCONFIRMED_SHIPPING_ADDRESS • UNILATERAL • VERIFICATION_REQUIRED  Note: The returned value can also include the processor response values. See Reason Codes (on page 103).		
apSaleReply_processorTransactionFee	Amount up to N digit after the decimal separator as defined in ISO 4217 for the appropriate currency code.	Sale	String (15)
apSaleReply_processorTransactionID	Identifier of the order transaction. Read-only.	Sale	String (50)
apSaleReply_reasonCode	Numeric value corresponding to the result of the sale service request. See Reason Codes (on page 103) .	Sale	Integer (5)
apSaleReply_reconciliationID	Reference number that you use to reconcile your Cybersource reports with your reports.	Sale	String (60)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
apSessionsReply_amount	Total amount charged from the payer to the payee. In the case of a refund, this is the refunded amount to the original payer from the payee. Maximum of 10 characters with support for 2 decimal places.	Sessions	String (15)
apSessionsReply_dateTime	Time of session. Format: yyyy-MM-DDThh:mm:ssZ 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	Sessions	String (18)
apSessionsReply_merchantURL	Redirect URL to PayPal web site.	Sessions	String (2048)
apSessionsReply_paymentStatus	Status of the payment request. Possible values: • CREATED • FAILED	Sessions	String (15)
apSessionsReply_processorTransactionID	ID of the created payment. Value assigned by PayPal.	Sessions	String (50)
apSessionsReply_reasonCode	Numeric value corresponding to the result of the sessions service request.	Sessions	Integer (5)
apSessionsReply_reconciliationID	Invoice number to track this payment. Maximum length: 127.	Sessions	String(127)
apSessionsReply_responseCode	The transaction response code received from the processor.	Sessions	String (5)
apSessionsReply_status	Status of the sessions request.	Sessions	String (15)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	Possible values: • CREATED • FAILED		
billTo_city	City of the billing address.  Note: To get the details of a customer's billing address from PayPal using the check status service, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.	Billing Agreement Check Status Order	String (50)
billTo_country	Country of the billing address. Use the two-character ISO Standard Country Codes.  Note: To get the details of a customer's billing address from Paypal using the check status service, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.	Billing Agreement Check Status Order	String (2)
billTo_email	Customer's email address.  Note: To get the details of a customer's billing address from PayPal using the check status service, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.	Billing Agreement Check Status Order	String (256)
billTo_firstName	First name of customer.	Billing Agreement	String (60)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	 Note: To get the details of a customer's billing address from PayPal using the check status service, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.	Check Status Order	
billTo_lastName	Last name of customer.  Note: To get the details of a customer's billing address from PayPal using the check status service, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.	Billing Agreement Check Status Order	String (60)
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] 12345-6789 When the billing country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space][numeric][alpha][numeric] A1B 2C3  Note: To get the details of a customer's billing address from PayPal using the check status service,	Billing Agreement Check Status Order	String (10)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	you must have this feature enabled on your PayPal account. Contact your PayPal account manager.		
billTo_state	State or province of the billing address. Use the State, Province, and Territory Codes for the United States and Canada.  Note: To get the details of a customer's billing address from PayPal using the check status service, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.	Billing Agreement Check Status Order	String (2)
billTo_street1	First line of the billing street address.  Note: To get the details of a customer's billing address from PayPal using the check status service, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.	Billing Agreement Check Status Order	String (60)
billTo_street2	Second line of the billing street address.  Note: To get the details of a customer's billing address from PayPal using the check status service, you must have this feature enabled on your PayPal account. Contact your PayPal account manager.	Billing Agreement Check Status Order	String (60)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
customerVerificationStatus	Possible values are VERIFIED or UNVERIFIED , depending on whether the buyer has verified their identity. Used in PayPal transactions.	Check Status Order	String (60)
decision	Summarizes the result of the overall request. For information about handling replies, see Getting Started with Cybersource Advanced for the Simple Order API . Possible values: • ACCEPT • ERROR • REJECT • REVIEW	All services	String (6)
invalidField_0 through invalidField_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 .	Authorization Auth Reversal Billing Agreement Cancel Capture Check Status Order Refund Sale Sessions	String (100)
reasonCode	Numeric value corresponding to the result of the overall request.	All services	Integer (5)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
	See Reason Codes (on page 103) .		
requestID	Identifier for the request generated by the client.	All services	String (26)
shipTo_city	City in the shipping address.	Billing Agreement Check Status Order	String (40)
shipTo_country	Country in the shipping address. This is a two-character ISO Standard Country Codes .	Billing Agreement Check Status Order	String (2)
shipTo_firstName	First name of recipient.	Billing Agreement Check Status Order	String (60)
shipTo_lastName	Last name of the recipient.	Billing Agreement Check Status Order	String (60)
shipTo_phoneNumber	Phone number of the recipient.	Billing Agreement Check Status Order	String (20)
shipTo_postalCode	Postal code of shipping address. Consists of 5 to 9 digits.	Billing Agreement Check Status Order	String (20)
shipTo_state	State or province in the shipping address. This is a State, Province, and Territory Codes for the United States and Canada .	Billing Agreement Check Status Order	String (40)

Response Fields (continued)

Field	Description	Returned By	Data Type & Length
shipTo_street1	First line of the shipping address.	Billing Agreement Check Status Order	String (100)
shipTo_street2	Second line of the shipping address.	Billing Agreement Check Status Order	String (100)

Service Examples

Name-Value Pair Examples

Sessions Service

Sessions Service Request

```
merchantID=mid123
merchantReferenceCode=1234
invoiceHeader_merchantDescriptor=MerchantDesc
shipTo_firstName=Jane
shipTo_lastName=Smith
shipTo_street1=123 Main Street
shipTo_street2=Mailstop A
shipTo_city=Small Town
shipTo_state=CA
shipTo_postalCode=98765
shipTo_country=US
shipTo_phoneNumber=9876543210
shipTo_shippingMethod=FedEx
item_0_unitPrice=45.00
item_0_quantity=2
item_0_productCode=default
item_0_productName=TestProduct
item_0_productSKU=TestSKU
item_0_productDescription=TestDesc
item_0_unitTaxAmount=1.00
purchaseTotals_currency=USD
purchaseTotals_taxAmount=2.00
purchaseTotals_grandTotalAmount=95.00
purchaseTotals_subtotalAmount=90.00
purchaseTotals_shippingAmount=5.00
purchaseTotals_handlingAmount=3.00
purchaseTotals_shippingDiscountAmount=5.00
apPaymentType=PPL
apSessionsService_run=true
apSessionsService_cancelURL=http://www.example.com
apSessionsService_successURL=http://www.example.com
```

Sessions Service Response

```
merchantReferenceCode=1234
requestID=4980396424596224104012
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=USD
apSessionsReply_reasonCode=100
apSessionsReply_merchantURL=https://www.sandbox.paypal.com/cgi-bin/webscr?
cmd=_express-checkout&token=EC-1YN71306R29977435
apSessionsReply_processorTransactionID=PAY-7PN460756LFFEKWY
apSessionsReply_amount=95.00
apSessionsReply_reconciliationID=40HXQXLD0000J3XKJCQV
apSessionsReply_status=CREATED
```

Order Service

Order Service Request

```
merchantID=mid123
merchantReferenceCode=1234
item_0_unitPrice=45.00
item_0_quantity=2
item_0_productName=TestProduct
item_0_productCode=default
item_0_productSKU=TestSKU
item_0_productDescription=TestDesc
item_0_unitTaxAmount=1.00
purchaseTotals_currency=USD
purchaseTotals_taxAmount=2.00
purchaseTotals_grandTotalAmount=95.00
purchaseTotals_subtotalAmount=90.00
purchaseTotals_shippingAmount=5.00
purchaseTotals_handlingAmount=3.00
purchaseTotals_shippingDiscountAmount=5.00
apPaymentType=PPL
ap_payerID=7FUDAXBNR5KSC
apOrderService_run=true
apOrderService_sessionsRequestID=4980396424596224104012
```

Order Service Response

```
merchantReferenceCode=1234
requestID=4980405898776291704008
```

```
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=USD
apReply_orderID=0-1CM486978D015523P
apReply_payerID=7FUDAXBNR5KSC
shipTo_firstName=Jane
shipTo_lastName=Smith
shipTo_street1=123 Main Street
shipTo_street2=Mailstop A
shipTo_city=Small Town
shipTo_state=CA
shipTo_postalCode=98765
shipTo_country=US
billTo_firstName=Jane
billTo_lastName=Smith
billTo_country=US
billTo_email=jsmith@example.com
apOrderReply_reasonCode=100
apOrderReply_amount=100.00
apOrderReply_reconciliationID=40HXQXLD0000J3XKJCQV
apOrderReply_status=CREATED
apOrderReply_processorResponse=ORDER
```

Cancel Service

Cancel Service Request

```
merchantID=mid123
merchantReferenceCode=1234
apPaymentType=PPL
apCancelService_run=true
apCancelService_orderRequestID=1124439821106016799952
```

Cancel Service Response

```
merchantReferenceCode=1234
requestID=3333330307786014000000
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=USD
apCancelReply_reasonCode=100
apCancelReply_processorTransactionID=07W241641VD265833L
apCancelReply_status=CANCELLED
```

Authorization Service

Authorization Service Request

```
merchantID=mid123
merchantReferenceCode=1234
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=95.00
apPaymentType=PPL
apAuthService_run=true
apAuthService_orderRequestID=4980405898776291704008
```

Authorization Service Response

```
merchantReferenceCode=1234
requestID=4896155375886014701200
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=USD
apAuthReply_reasonCode=100
apAuthReply_transactionID=4KP97418B91840630
apAuthReply_amount=95.00
apAuthReply_paymentStatus=AUTHORIZED
apAuthReply_reconciliationID=ZZEWULIG00000000000000000000IZWESA00
apAuthReply_processorTransactionID=4KP97418B91840630
```

Authorization Reversal Service

Authorization Reversal Service Request

```
merchantID=mid123
merchantReferenceCode=1234
apPaymentType=PPL
apAuthReversalService_run=true
apAuthReversalService_AuthRequestID=4896155375886014701200
```

Authorization Reversal Service Response

```
merchantReferenceCode=1234
requestID=4896156883666666801208
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=USD
```

```
apAuthReversalReply_reasonCode=100
apAuthReversalReply_transactionID=4KP97418B91840630
apAuthReversalReply_amount=95.00
apAuthReversalReply_paymentStatus=AUTH_REVERSED
apAuthReversalReply_reconciliationID=ZZEWULIG0000000000IZWESA01
```

Capture Service

Capture Service Request

```
merchantID=mid123
merchantReferenceCode=1234
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=95.00
apPaymentType=PPL
apCaptureService_run=true
apCaptureService_AuthRequestID=4896155375886014701200
```

Capture Service Response

```
merchantReferenceCode=1234
requestID=6886190757355555601201
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=USD
apCaptureReply_reasonCode=100
apCaptureReply_transactionID=30M24208823051238
apCaptureReply_processorResponse=None
apCaptureReply_amount=95.00
apCaptureReply_reconciliationID=ZZEWULIG0000000000ZWESA0B
apCaptureReply_paymentStatus=SETTLED
apCaptureReply_responseCode=None
```

Sale Service

Sale Service Request

```
merchantID=mid123
merchantReferenceCode=1234
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=95.00
apPaymentType=PPL
```

```
apSaleService_run=true
apSaleService_orderRequestID=6886190757355555601201
```

Sale Service Response

```
merchantReferenceCode=1234
requestID=1124439821106016799952
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=USD
apSaleReply_reasonCode=100
apSaleReply_paymentStatus=SETTLED
apSaleReply_responseCode=None
apSaleReply_processorTransactionID=8HC63549VB1392407
apSaleReply_reconciliationID=ZZEWULIG0000000000IZWESAOF
apSaleReply_processorTransactionFee=3.20
apSaleReply_amount=95.00
apSaleReply_processorResponse=None
```

Sale Service Request (for Reference Transaction)

```
merchantID=mid123
merchantReferenceCode=1234
invoiceHeader_merchantDescriptor=MerchantDesc
shipTo_firstName=Jane
shipTo_lastName=Smith
shipTo_street1=123 Main Street
shipTo_street2=Mailstop A
shipTo_city=Small Town
shipTo_state=CA
shipTo_postalCode=98765
shipTo_country=US
shipTo_phoneNumber=9876543210
shipTo_shippingMethod=FedEx
item_0_unitPrice=45.00
item_0_quantity=2
item_0_productCode=default
item_0_productName=TestProduct
item_0_productSKU=TestSKU
purchaseTotals_currency=USD
purchaseTotals_taxAmount=2.00
purchaseTotals_grandTotalAmount=95.00
purchaseTotals_subtotalAmount=90.00
purchaseTotals_shippingAmount=5.00
purchaseTotals_handlingAmount=3.00
purchaseTotals_shippingDiscountAmount=5.00
apPaymentType=PPL
```

```
ap_billingAgreementID=SOK-3FJ6025
ap_fundingSource=unrestricted
apSaleService_run=true
apSaleService_orderRequestID=6886190757355555601201
```

Sale Service Response (for Reference Transaction)

```
merchantReferenceCode=1234
requestID=4980405898776291709999
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=USD
apReply_sellerProtection_eligibility=ELIGIBLE
apReply_sellerProtection_type=ITEM_NOT_RECEIVED_ELIGIBLE
apReply_processorFraudDecision=ACCEPT
apReply_processorFraudDecisionReason=AVS_NO_MATCH
apReply_billingAgreementID=B-WE812486D0938631
apReply_payerID=3N53BA8B3MCU4
apReply_fundingSource=INSTANT_TRANSFER
apSaleReply_reasonCode=100
apSaleReply_paymentStatus=SETTLED
apSaleReply_processorTransactionID=SOK-3FJ6025
apSaleReply_reconciliationID=40HXQXLD0000J3XKJCQV
apSaleReply_processorTransactionFee=1.05
apSaleReply_amount=100.00
apSaleReply_dateTime=2019-08-11T01:45:39Z
```

Refund Service

Refund Service Request

```
merchantID=mid123
merchantReferenceCode=1234
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=95.00
apPaymentType=PPL
apRefundService_run=true
apRefundService_refundRequestID=1124439821106016799952
```

Refund Service Response

```
merchantReferenceCode=1234
requestID=48961921504567098712009999
decision=ACCEPT
```

```
reasonCode=100
purchaseTotals_currency=USD
apRefundReply_reasonCode=100
apRefundReply_amount=95.00
apRefundReply_reconciliationID=ZZEWULIG0000000000IZWESAOC
apRefundReply_processorTransactionID=8FU4435670486773H
apRefundReply_paymentStatus=REFUNDED
```

Billing Agreement Service

Billing Agreement Service Request

```
merchantID=mid123
merchantReferenceCode=1234
apPaymentType=PPL
apBillingAgreementService_run=true
apBillingAgreementService_sessionsRequestID=4980396424596224104012
```

Billing Agreement Service Response

```
merchantReferenceCode-1234
requestID=4955835535216022201540
decision=ACCEPT
reasonCode=100
apReply_billingAgreementID=S0K3FJ6025
shipTo_firstName=Jane
shipTo_lastName=Smith
shipTo_street1=123 Main St
shipTo_street2=Mailstop A
shipTo_city=Small Town
shipTo_state=CA
shipTo_postalCode=98765
shipTo_country=US
shipTo_phoneNumber=9876543210
billTo_firstName=Jane
billTo_lastName=Smith
billTo_street1=123 Main Street
billTo_street2=Mailstop A
billTo_city=Small Town
billTo_state=CA
billTo_postalCode=98765
billTo_country=US
billTo_phoneNumber=9876543210
billTo_email=jsmith@example.com
apBillingAgreementReply_reasonCode=100
```

XML Examples

Sessions Service

Sessions Service Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <invoiceHeader>
    <merchantDescriptor>MerchantDesc</merchantDescriptor>
  </invoiceHeader>
  <shipTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <street2>Mailstop A</street2>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <phoneNumber>9876543210</phoneNumber>
    <shippingMethod>FedEx</shippingMethod>
  </shipTo>
  <item id="0">
    <unitPrice>45.00</unitPrice>
    <quantity>2</quantity>
    <productCode>default</productCode>
    <productName>TestProduct</productName>
    <productSKU>TestSKU</productSKU>
    <productDescription>TestDesc</productDescription>
    <unitTaxAmount>1.00</unitTaxAmount>
  </item>
  <purchaseTotals>
    <currency>USD</currency>
    <taxAmount>2.00</taxAmount>
    <grandTotalAmount>95.00</grandTotalAmount>
    <subtotalAmount>90.00</subtotalAmount>
    <shippingAmount>5.00</shippingAmount>
    <handlingAmount>3.00</handlingAmount>
```

```

        <shippingDiscountAmount>5.00</shippingDiscountAmount>
    </purchaseTotals>
    <apPaymentType>PPL</apPaymentType>
    <apSessionsService run="true">
        <cancelURL>http://www.example.com</cancelURL>
        <successURL>http://www.example.com</successURL>
    </apSessionsService>
</requestMessage>

```

Sessions Service Response

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
    <c:merchantReferenceCode>1234</c:merchantReferenceCode>
    <c:requestID>4980396424596224104012</c:requestID>
    <c:decision>ACCEPT</c:decision>
    <c:reasonCode>100</c:reasonCode>
    <c:purchaseTotals>
        <c:currency>USD</c:currency>
    </c:purchaseTotals>
    <c:apSessionsReply>
        <c:reasonCode>100</c:reasonCode>
        <c:merchantURL>https://www.sandbox.paypal.com/cgi-bin/webscr?
cmd=_express-checkout&token=EC-1YN71306R29977435</c:merchantURL>
        <c:processorTransactionID>PAY-7PN460756LFFEKWY</c:processorTransactionID>
        <c:amount>95.00</c:amount>
        <c:reconciliationID>40HXQXLD0000J3XKJCQV</c:reconciliationID>
    <c:status>CREATED</c:status>
    </c:apSessionsReply>
</c:replyMessage>

```

Order Service

Order Service Request

```

<requestMessage xmlns="urn:schemas--com:transaction-data-1.161">
    <merchantID>mid123</merchantID>
    <merchantReferenceCode>1234</merchantReferenceCode>
    <item id="0">
        <unitPrice>45.00</unitPrice>
        <quantity>2</quantity>
        <productName>TestProduct</productName>
        <productCode>default</productCode>
        <productSKU>TestSKU</productSKU>
        <productDescription>TestDesc</productDescription>
        <unitTaxAmount>1.00</unitTaxAmount>
    </item>

```

```

</item>
<purchaseTotals>
  <currency>USD</currency>
  <taxAmount>2.00</taxAmount>
  <grandTotalAmount>95.00</grandTotalAmount>
  <subtotalAmount>90.00</subtotalAmount>
  <shippingAmount>5.00</shippingAmount>
  <handlingAmount>3.00</handlingAmount>
  <shippingDiscountAmount>5.00</shippingDiscountAmount>
</purchaseTotals>
<apPaymentType>PPL</apPaymentType>
<ap>
  <payerID>7FUDAXBNR5KSC</payerID>
</ap>
<apOrderService run="true">
  <sessionsRequestID>4980396424596224104012</sessionsRequestID>
</apOrderService>
</requestMessage>

```

Order Service Response

```

<c:replyMessage xmlns:c="urn:schemas--com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>4980405898776291704008</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:apReply>
    <c:orderID>0-1CM486978D015523P</c:orderID>
    <c:payerID>7FUDAXBNR5KSC</c:payerID>
  </c:apReply>
  <c:shipTo>
    <c:firstName>Jane</c:firstName>
    <c:lastName>Smith</c:lastName>
    <c:street1>123 Main Street</c:street1>
    <c:street2>Mailstop A</c:street2>
    <c:city>Small Town</c:city>
    <c:state>CA</c:state>
    <c:postalCode>98765</c:postalCode>
    <c:country>US</c:country>
  </c:shipTo>
  <c:billTo>
    <c:firstName>Jane</c:firstName>
    <c:lastName>Smith</c:lastName>
    <c:country>US</c:country>
    <c:email>jsmith@example.com</c:email>

```

```

</c:billTo>
<c:apOrderReply>
  <c:reasonCode>100</c:reasonCode>
  <c:amount>100.00</c:amount>
  <c:reconciliationID>40HXQXLD0000J3XKJCQV</c:reconciliationID>
  <c:status>CREATED</c:status>
  <c:processorResponse>ORDER</c:processorResponse>
</c:apOrderReply>
</c:replyMessage>

```

Cancel Service

Cancel Service Request

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <apPaymentType>PPL</apPaymentType>
  <apCancelService run="true">
    <orderRequestID>1124439821106016799952</orderRequestID>
  </apCancelService>
</requestMessage>

```

Cancel Service Response

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>3333330307786014000000</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:apCancelReply>
    <c:reasonCode>100</c:reasonCode>
    <c:processorTransactionID>07W241641VD265833L</c:processorTransactionID>
    <c:status>CANCELLED</c:status>
  </c:apCancelReply>
</c:replyMessage>

```

Authorization Service

Authorization Service Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>95.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>PPL</apPaymentType>
  <apAuthService run="true">
    <orderRequestID>4980405898776291704008</orderRequestID>
  </apAuthService>
</requestMessage>
```

Authorization Service Response

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>4896155375886014701200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:apAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:transactionID>4KP97418B91840630</c:transactionID>
    <c:amount>95.00</c:amount>
    <c:paymentStatus>AUTHORIZED</c:paymentStatus>
    <c:reconciliationID>ZZEWULIG000000000000IZWESA00</c:reconciliationID>
    <c:processorTransactionID>4KP97418B91840630</c:processorTransactionID>
  </c:apAuthReply>
</c:replyMessage>
```

Authorization Reversal Service

Authorization Reversal Service Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
```

```
<apPaymentType>PPL</apPaymentType>
<apAuthReversalService run="true">
  <authRequestID>4896155375886014701200</authRequestID>
</apAuthReversalService>
</requestMessage>
```

Authorization Reversal Service Response

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>4896156883666666801208</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:apAuthReversalReply>
    <c:reasonCode>100</c:reasonCode>
    <c:transactionID>4KP97418B91840630</c:transactionID>
    <c:amount>95.00</c:amount>
    <c:paymentStatus>AUTH_REVERSED</c:paymentStatus>
    <c:reconciliationID>ZZEWULIG00000000000000000000IZWESA01</c:reconciliationID>
  </c:apAuthReversalReply>
</c:replyMessage>
```

Capture Service

Capture Service Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>95.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>PPL</apPaymentType>
  <apCaptureService run="true">
    <authRequestID>4896155375886014701200</authRequestID>
  </apCaptureService>
</requestMessage>
```

Capture Service Response

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>6886190757355555601201</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:apCaptureReply>
    <c:reasonCode>100</c:reasonCode>
    <c:processorTransactionID>30M24208823051238</c:processorTransactionID>
    <c:processorResponse>None</c:processorResponse>
    <c:amount>95.00</c:amount>
    <c:reconciliationID>ZZEWULIG00000000000000000000IZWESA0B</c:reconciliationID>
    <c:paymentStatus>SETTLED</c:paymentStatus>
    <c:responseCode>None</c:responseCode>
  </c:apCaptureReply>
</c:replyMessage>

```

Sale Service

Sale Service Request

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>95.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>PPL</apPaymentType>
  <apSaleService run="true">
    <orderRequestID>6886190757355555601201</orderRequestID>
  </apSaleService>
</requestMessage>

```

Sale Service Response

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>1124439821106016799952</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>

```

```

    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:apSaleReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>SETTLED</c:paymentStatus>
    <c:responseCode>None</c:responseCode>
    <c:processorTransactionID>8HC63549VB1392407</c:processorTransactionID>
    <c:reconciliationID>ZZEWULIG0000000000IZWESA0F</c:reconciliationID>
    <c:processorTransactionFee>3.20</c:processorTransactionFee>
    <c:amount>95.00</c:amount>
    <c:processorResponse>None</c:processorResponse>
  </c:apSaleReply>
</c:replyMessage>

```

Sale Service Request (for Reference Transaction)

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <invoiceHeader>
    <merchantDescriptor>MerchantDesc</merchantDescriptor>
  </invoiceHeader>
  <shipTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <street2>Mailstop A</street2>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <phoneNumber>9876543210</phoneNumber>
    <shippingMethod>FedEx</shippingMethod>
  </shipTo>
  <item id="0">
    <unitPrice>45.00</unitPrice>
    <quantity>2</quantity>
    <productCode>default</productCode>
    <productName>TestProduct</productName>
    <productSKU>TestSKU</productSKU>
  </item>
  <purchaseTotals>
    <currency>USD</currency>
    <taxAmount>2.00</taxAmount>
    <grandTotalAmount>95.00</grandTotalAmount>
    <subtotalAmount>90.00</subtotalAmount>
    <shippingAmount>5.00</shippingAmount>
    <handlingAmount>3.00</handlingAmount>
  </purchaseTotals>
</requestMessage>

```

```

        <shippingDiscountAmount>5.00</shippingDiscountAmount>
    </purchaseTotals>
    <apPaymentType>PPL</apPaymentType>
    <ap>
        <billingAgreementID>SOK-3FJ6025</billingAgreementID>
        <fundingSource>unrestricted</fundingSource>
    </ap>
    <apSaleService run="true">
        <orderRequestID>6886190757355555601201</orderRequestID>
    </apSaleService>
</requestMessage>

```

Sale Service Response (for Reference Transaction)

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
    <c:merchantReferenceCode>1234</c:merchantReferenceCode>
    <c:requestID>4980405898776291709999</c:requestID>
    <c:decision>ACCEPT</c:decision>
    <c:reasonCode>100</c:reasonCode>
    <c:purchaseTotals>
        <c:currency>USD</c:currency>
    </c:purchaseTotals>
    <c:apReply>
        <c:sellerProtection>
            <c:eligibility>ELIGIBLE</c:eligibility>
            <c:type>ITEM_NOT_RECEIVED_ELIGIBLE</c:type>
        </c:sellerProtection>
        <c:processorFraudDecision>ACCEPT</c:processorFraudDecision>

        <c:processorFraudDecisionReason>AVS_NO_MATCH</c:processorFraudDecisionReason>
        <c:billingAgreementID>B-WE812486D0938631</c:billingAgreementID>
        <c:payerID>3N53BA8B3MCU4</c:payerID>
        <c:fundingSource>INSTANT_TRANSFER</c:fundingSource>
    </c:apReply>
    <c:apSaleReply>
        <c:reasonCode>100</c:reasonCode>
        <c:paymentStatus>SETTLED</c:paymentStatus>
        <c:processorTransactionID>SOK-3FJ6025</c:processorTransaction>
        <c:reconciliationID>40HXQXLD0000J3XKJCQV</c:reconciliationID>
        <c:processorTransactionFee>1.05</c:processorTransactionFee>
        <c:amount>100.00</c:amount>
        <c:dateTime>2019-08-11T01:45:39Z</c:dateTime>
    </c:apSaleReply>
</c:replyMessage>

```

Refund Service

Refund Service Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>95.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>PPL</apPaymentType>
  <apRefundService run="true">
    <refundRequestID>1124439821106016799952</refundRequestID>
  </apRefundService>
</requestMessage>
```

Refund Service Response

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>48961921504567098712009999</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:apRefundReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>95.00</c:amount>
    <c:reconciliationID>ZZEWULIG0000000000IZWESAOC</c:reconciliationID>
    <c:processorTransactionID>8FU4435670486773H</c:processorTransactionID>
    <c:paymentStatus>REFUNDED</c:paymentStatus>
  </c:apRefundReply>
</c:replyMessage>
```

Billing Agreement Service

Billing Agreement Service Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <apPaymentType>PPL</apPaymentType>
```

```
<apBillingAgreementService run="true">
  <sessionsRequestID>4980396424596224104012</sessionsRequestID>
</apBillingAgreementService>
</requestMessage>
```

Billing Agreement Service Response

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>4955835535216022201540</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:apReply>
    <c:billingAgreementID>SOK3FJ6025</c:billingAgreementID>
  </c:apReply>
  <c:shipTo>
    <c:firstName>Jane</c:firstName>
    <c:lastName>Smith</c:lastName>
    <c:street1>123 Main St</c:street1>
    <c:street2>Mailstop A</c:street2>
    <c:city>Small Town</c:city>
    <c:state>CA</c:state>
    <c:postalCode>98765</c:postalCode>
    <c:country>US</c:country>
    <c:phoneNumber>9876543210</c:phoneNumber>
  </c:shipTo>
  <c:billTo>
    <c:firstName>Jane</c:firstName>
    <c:lastName>Smith</c:lastName>
    <c:street1>123 Main Street</c:street1>
    <c:street2>Mailstop A</c:street2>
    <c:city>Small Town</c:city>
    <c:state>CA</c:state>
    <c:postalCode>98765</c:postalCode>
    <c:country>US</c:country>
    <c:phoneNumber>9876543210</c:phoneNumber>
    <c:email>jsmith@example.com</c:email>
  </c:billTo>
  <c:apBillingAgreementReply>
    <c:reasonCode>100</c:reasonCode>
    <c:status>ACTIVE</c:status>
  </c:apBillingAgreementReply>
</c:replyMessage>
```



```
billTo_postalCode=98765  
billTo_email=jsmith@example.com  
customerVerificationStatus=UNVERIFIED
```

Order Service: Check Status

Check Status Request: Order Service

```
merchantID=mid123  
merchantReferenceCode=1234  
apPaymentType=PPL  
apCheckStatusService_run=true  
apCheckStatusService_checkStatusRequestID=4980405898776291704008
```

Check Status Response: Order Service

```
merchantReferenceCode=1234  
requestID=4896415760916002201200  
decision>ACCEPT  
reasonCode=100  
apCheckStatusReply_reasonCode=100  
apCheckStatusReply_paymentStatus=CREATED  
apCheckStatusReply_processorTransactionID=05TD10988AD299315S
```

Authorization Service: Check Status

Check Status Request: Authorizations Service

```
merchantID=mid123  
merchantReferenceCode=1234  
apPaymentType=PPL  
apCheckStatusService_run=true  
apCheckStatusService_checkStatusRequestID=4896155375886014701200
```

Check Status Response: Authorization Service

```
merchantReferenceCode=1234  
requestID=4896452294406002901200  
decision=ACCEPT  
reasonCode=100  
apCheckStatusReply_reasonCode=100
```

```
apCheckStatusReply_paymentStatus=AUTHORIZED
apCheckStatusReply_processorTransactionID=4KP97418B91840630
```

Capture Service: Check Status

Check Status Request: Capture Service

```
merchantID=mid123
merchantReferenceCode=1234
apPaymentType=PPL
apCheckStatusService_run=true
apCheckStatusService_checkStatusRequestID=688619075735555601201
```

Check Status Response: Capture Service

```
merchantReferenceCode=1234
requestID=7432196931186003001299
decision=ACCEPT
reasonCode=100
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID>ZZEWULIG00000000000000000000IZWESA07
apCheckStatusReply_paymentStatus=SETTLED
apCheckStatusReply_processorTransactionID=30M24208823051238
```

Sale Service: Check Status

Check Status Request: Sale Service

```
merchantID=mid123
merchantReferenceCode=1234
apPaymentType=PPL
apCheckStatusService_run=true
apCheckStatusService_checkStatusRequestID=1124439821106016799952
```

Check Status Response: Sale Service

```
merchantReferenceCode=1234
requestID=500722420177946002301009
decision=ACCEPT
reasonCode=100
apCheckStatusReply_reasonCode=100
```

```
apCheckStatusReply_reconciliationID=0ELJXCHZ0000000000000000000J0BLF9SV
apCheckStatusReply_paymentStatus=SETTLED
apCheckStatusReply_processorTransactionID=7LE62945SB636124F
```

Refund Service: Check Status

Check Status Request: Refund Service

```
merchantID=mid123
merchantReferenceCode=1234
apPaymentType=PPL
apCheckStatusService_run=true
apCheckStatusService_checkStatusRequestID=48961921504567098712009999
```

Check Status Response: Refund Service

```
merchantReferenceCode=1234
requestID=7649816002777888
decision=ACCEPT
reasonCode=100
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=ZZEWULIG00000000000000000000IZWESAOC
apCheckStatusReply_paymentStatus=REFUNDED
apCheckStatusReply_processorTransactionID=8FU4435670486773H
```

Billing Agreement Service: Check Status

Check Status Request: Billing Agreement Service

```
merchantID=mid123
merchantReferenceCode=1234
apPaymentType=PPL
apCheckStatusService_run=true
apCheckStatusService_sessionsRequestID=4955835535216022201540
ap_billingAgreementID=S0K3FJ6025
```

Check Status Response: Billing Agreement Service

```
merchantReferenceCode=1234
requestID=4926430141876011901200
decision=ACCEPT
```

```
reasonCode=100
apCheckStatusReply_reasonCode=100
apCheckStatusReply_paymentStatus=ACTIVE
apCheckStatusReply_processorTransactionID=B4DR69719PY986990T
apReply_payerID=3N53BA8B3MCU4
shipTo_firstName=Jane
shipTo_lastName=Smith
shipTo_street1=123 Main Street
shipTo_street2=Mailstop A
shipTo_city=Small Town
shipTo_state=CA
shipTo_country=US
shipTo_postalCode=98765
billTo_firstName=Jane
billTo_lastName=Smith
billTo_street1=123 Main St
billTo_city=Small Town
billTo_state=CA
billTo_country=US
billTo_postalCode=98765
billTo_email=jsmith@example.com
```

XML Examples

Sessions Service: Check Status

Check Status Request: Sessions Service

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <apPaymentType>PPL</apPaymentType>
  <apCheckStatusService run="true">
    <sessionsRequestID>4980396424596224104012</sessionsRequestID>
  </apCheckStatusService>
</requestMessage>
```

Check Status Response: Sessions Service

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
```

```

<c:requestID>4980396424596224104012</c:requestID>
<c:decision>ACCEPT</c:decision>
<c:reasonCode>100</c:reasonCode>
<c:apCheckStatusReply>
  <c:reasonCode>100</c:reasonCode>
  <c:reconciliationID>ZZEWULIG000000000000000000000000IZWESAOG</c:reconciliationID>
  <c:paymentStatus>CREATED</c:paymentStatus>

<c:processorTransactionID>PAY-78U923845790YLDE5KZY</c:processorTransactionID>
</c:apCheckStatusReply>
<c:apReply>
  <c:payerID>7FUDAXBNR5KSC</c:payerID>
</c:apReply>
<c:shipTo>
  <c:firstName>Jane</c:firstName>
  <c:lastName>Smith</c:lastName>
  <c:street1>123 Main Street</c:street1>
  <c:street2>Mailstop A</c:street2>
  <c:city>Small Town</c:city>
  <c:state>CA</c:state>
  <c:country>US</c:country>
  <c:phoneNumber>9876543210</c:phoneNumber>
  <c:postalCode>98765</c:postalCode>
</c:shipTo>
<c:billTo>
  <c:firstName>Jane</c:firstName>
  <c:lastName>Smith</c:lastName>
  <c:street1>123 Main St</c:street1>
  <c:city>Small Town</c:city>
  <c:state>CA</c:state>
  <c:country>US</c:country>
  <c:postalCode>98765</c:postalCode>
  <c:email>jsmith@example.com</c:email>
</c:billTo>
<c:customerVerificationStatus>UNVERIFIED</c:customerVerificationStatus>
</c:replyMessage>

```

Order Service: Check Status

Check Status Request: Order Service

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <apPaymentType>PPL</apPaymentType>
  <apCheckStatusService run="true">

```

```
<checkStatusRequestID>4980405898776291704008</checkStatusRequestID>
</apCheckStatusService>
</requestMessage>
```

Check Status Response: Order Service

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>4896415760916002201200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>CREATED</c:paymentStatus>
    <c:processorTransactionID>05TD10988AD299315S</c:processorTransactionID>
  </c:apCheckStatusReply>
</c:replyMessage>
```

Authorization Service: Check Status

Check Status Request: Authorization Service

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <apPaymentType>PPL</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>4896155375886014701200</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

Check Status Response: Authorization Service

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>4896452294406002901200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>AUTHORIZED</c:paymentStatus>
    <c:processorTransactionID>4KP97418B91840630</c:processorTransactionID>
  </c:apCheckStatusReply>
</c:replyMessage>
```

Capture Service: Check Status

Check Status Request: Capture Service

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <apPaymentType>PPL</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>6886190757355555601201</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

Check Status Response: Capture Service

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>7432196931186003001299</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:reconciliationID>ZZEWULIG000000000000000000000000IZWESA07</c:reconciliationID>
    <c:paymentStatus>SETTLED</c:paymentStatus>
    <c:processorTransactionID>30M24208823051238</c:processorTransactionID>
  </c:apCheckStatusReply>
</c:replyMessage>
```

Sale Service: Check Status

Check Status Request: Sale Service

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <apPaymentType>PPL</apPaymentType>
  <apCheckStatusService run="true">
    <CheckStatusRequestID>1124439821106016799952</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

Check Status Response: Sale Service

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>500722420177946002301009</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:reconciliationID>0ELJXCHZ00000000000000000000J0BLF9SV</c:reconciliationID>
    <c:paymentStatus>SETTLED</c:paymentStatus>
    <c:processorTransactionID>7LE62945SB636124F</c:processorTransactionID>
  </c:apCheckStatusReply>
</c:replyMessage>

```

Refund Service: Check Status

Check Status Request: Refund Service

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <apPaymentType>PPL</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>48961921504567098712009999</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>

```

Check Status Response: Refund Service

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>7649816002777888</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:reconciliationID>ZZEWULIG00000000000000000000IZWESAOC</c:reconciliationID>
    <c:paymentStatus>REFUNDED</c:paymentStatus>
    <c:processorTransactionID>8FU4435670486773H</c:processorTransactionID>
  </c:apCheckStatusReply>
</c:replyMessage>

```

Billing Agreement Service: Check Status

Check Status Request: Billing Agreement Service

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.161">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>1234</merchantReferenceCode>
  <apPaymentType>PPL</apPaymentType>
  <apCheckStatusService run="true"/>
    <sessionsRequestID>4955835535216022201540</sessionsRequestID>
  </ap>
  <billingAgreementID>S0K3FJ6025</billingAgreementID>
</ap>
</requestMessage>
```

Check Status Response: Billing Agreement Service

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.161">
  <c:merchantReferenceCode>1234</c:merchantReferenceCode>
  <c:requestID>4926430141876011901200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>ACTIVE</c:paymentStatus>
    <c:processorTransactionID>B4DR69719PY986990T</c:processorTransactionID>
  </c:apCheckStatusReply>
  <c:apReply>
    <c:payerID>3N53BA8B3MCU4</c:payerID>
  </c:apReply>
  <c:shipTo>
    <c:firstName>Jane</c:firstName>
    <c:lastName>Smith</c:lastName>
    <c:street1>123 Main Street</c:street1>
    <c:street2>Mailstop A</c:street2>
    <c:city>Small Town</c:city>
    <c:state>CA</c:state>
    <c:country>US</c:country>
    <c:postalCode>98765</c:postalCode>
  </c:shipTo>
  <c:billTo>
    <c:firstName>Jane</c:firstName>
    <c:lastName>Smith</c:lastName>
    <c:street1>123 Main St</c:street1>
    <c:city>Small Town</c:city>
    <c:state>CA</c:state>
    <c:country>US</c:country>
```

```
<c:postalCode>98765</c:postalCode>  
  <c:email>jsmith@example.com</c:email>  
</c:billTo>  
</c:replyMessage>
```

Reason Codes

The reason codes returned by the Simple Order API for the PayPal services are described in the table below. For a more detailed description of replies, decisions, and reason codes, and for information about handling responses, see [Getting Started with Cybersource Advanced for the Simple Order API](#).

 **Important:** Because Cybersource can add response fields and reason codes at any time:

You must parse the response data according to the names of the fields instead of the field order in the response. For more information about parsing response 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.

For a description of each processor response, see:

<https://developer.paypal.com/docs/api/payments/#errors>

Reason Codes and Processor Responses

Reason Code	Description	Processor Responses
100	Successful transaction.	
150	General system failure. See your Cybersource client documentation for information about handling retries in the case of system errors.	DATA_RETRIEVAL DUPLICATE_REQUEST_ID INTERNAL_SERVICE_ERROR PAYMENT_REQUEST_ID_INVALID
202	Expired card. You might also receive this value if the expiration date you provided does not match the date the issuing bank has on file. Possible action: Request a different card or other form of payment.	EXPIRED_CREDIT_CARD

Reason Codes and Processor Responses (continued)

Reason Code	Description	Processor Responses
203	General decline of the card. No other information was provided by the issuing bank. Possible action: Request a different card or other form of payment.	CREDIT_CARD_CVV_CHECK_FAILED CREDIT_CARD_REFUSED
223	Possible reasons: • Merchant account or customer's account is not set up to process such transactions. • Insufficient funds in the customer's funding source associated with the account, or transaction declined by bank. • A particular action is not permitted, for example: capture refused, or the authorization has already been captured. • Fraud setting for the seller is blocking such payments. Payment approval by the customer has expired, and the merchant must restart the payment flow starting from payment creation and direct the customer back to PayPal.  Important: If an order or authorization returns INSTRUMENT_DECLINED, direct the customer to their PayPal account to choose a different funding option and resubmit the order. Possible reasons for the INSTRUMENT_DECLINED response can include: • The billing address associated with the financial instrument could not be confirmed.	BUYER_NOT_SET CANNOT_PAY_SELF CANNOT_REAUTH_CHILD_AUTHORIZATION CANNOT_REAUTH_INSIDE_HONOR_PERIOD CREDIT_PAYMENT_NOT_ALLOWED EXPIRED_CREDIT_CARD_TOKEN FAILED_TO_CHARGE_CC FEATURE_UNSUPPORTED_FOR_PAYEE FULL_REFUND_NOT_ALLOWED_AFTER_PARTIAL_REFUND IMMEDIATE_PAY_NOT_SUPPORTED INSTRUMENT_DECLINED INSUFFICIENT_FUNDS INVALID_FACILITATOR_CONFIGURATION MAXIMUM_ALLOWED_AUTHORIZATION_REACHED_FOR_ORDER MERCHANT_NOT_ENABLED_FOR_CHANNEL_INITIATED_BILLING

Reason Codes and Processor Responses (continued)

Reason Code	Description	Processor Responses
	The transaction exceeds the card limit. The transaction was denied by the card issuer.	MERCHANT_NOT_ENABLED_FOR_REFERENCE_TRANSACTION NEED_CREDIT_CARD NEED_CREDIT_CARD_OR_BANK_ACCOUNT NOT_IMPLEMENTED NO_EXTERNAL_FUNDING_DETAILS_FOUND ORDER_ALREADY_COMPLETED ORDER_VOIDED PAYEE_ACCOUNT_LOCKED_OR_CLOSED PAYEE_ACCOUNT_NO_CONFIRMED_EMAIL PAYEE_ACCOUNT_RESTRICTED PAYEE_BLOCKED_TRANSACTION PAYEE_COUNTRY_NOT_ENABLED PAYER_ACCOUNT_RESTRICTED PAYER_ACTION_REQUIRED PAYER_AUTHENTICATION_REQUIRED PAYER_CANNOT_PAY PAYER_COUNTRY_NOT_ENABLED PAYMENT_ALREADY_DONE

Reason Codes and Processor Responses (continued)

Reason Code	Description	Processor Responses
		PAYMENT_APPROVAL_EXPIRED PAYMENT_CANNOT_BE_INITIATED PAYMENT_DENIED PAYMENT_EXPIRED PAYMENT_METHOD_UNUSABLE PAYMENT_NOT_APPROVED_FOR_EXECUTION PAYMENT_STATE_INVALID REDIRECT_PAYER_FOR_ALTERNATE_FUNDING REFUND_EXCEEDED_TRANSACTION_AMOUNT REFUND_TIME_LIMIT_EXCEEDED REQUIRED_SCOPE_MISSING SENDING_LIMIT_EXCEEDED TOO_MANY_REAUTHORIZATIONS TRANSACTION_ALREADY_REFUNDED TRANSACTION_LIMIT_EXCEEDED TRANSACTION_REFUSED TRANSACTION_REFUSED_BY_PAYPAL_RISK

Reason Codes and Processor Responses (continued)

Reason Code	Description	Processor Responses
		TRANSACTION_REFUSED_PAYEE_PREFERENCE UNSUPPORTED_PAYEE_COUNTRY UNSUPPORTED_PAYEE_CURRENCY UNSUPPORTED_SEPA_BANK
231	Invalid account number. Possible action: Request a different card or other form of payment.	BANK_ACCOUNT_VALIDATION_FAILED
233	General decline by the processor.	AMOUNT_MISMATCH AGREEMENT_ALREADY_CANCELLED BATOKEN_MISMATCH BANK_MRN_MISMATCH BUYER_COUNTRY_NOT_ENABLED CANNOT_PAY_THIS_PAYEE CURRENCY_MISMATCH CURRENCY_NOT_ALLOWED EXECUTE_AGREEMENT_BUYER_NOT_ACCEPTED EXECUTE_AGREEMENT_ALREADY_CREATED EXECUTE_AGREEMENT_DOES_NOT_OWN_TOKEN INVALID_ARGUMENT INVALID_CITY_STATE_ZIP

Reason Codes and Processor Responses (continued)

Reason Code	Description	Processor Responses
		INVALID_EXPERIENCE_PROFILE_ID INVALID_PAYER_ID INVALID_PAYMENT_METHOD MERCHANT_COUNTRY_NOT_ENABLED REFUSED_MARK_REF_TXN_NOT_ENABLED PAYEE_FILTER_RESTRICTIONS PAYER_CANNOT_PAY PAYER_INFO_NULL PAYER_EMPTY BILLING ADDRESS PAYER_ID_MISSING_FOR_CARD_TOKEN PHONE_NUMBER_REQUIRED REDIRECT_REQUIRED RT_AGREEMENT_ALREADY_CANCELED SELECTED_PLAN_NOT_AVAILABLE SHIPPING_ADDRESS_INVALID UNABLE_TO_COMPLETE_TRANSACTION
235	The requested capture amount exceeds the originally authorized amount. Possible action: Issue a new authorization and capture request for the new amount.	CAPTURE_AMOUNT_LIMIT_EXCEEDED

Reason Codes and Processor Responses (continued)

Reason Code	Description	Processor Responses
237	The authorization has already been reversed. Possible action: No action required.	AUTHORIZATION_VOIDED
242	You requested a capture, but there is no corresponding, unused authorization record. Occurs if there was not a previously successful authorization request or if the previously successful authorization has already been used by another capture request. Possible action: Request a new authorization, and if successful, proceed with the capture.	AUTHORIZATION_EXPIRED AUTHORIZATION_ID_DOES_NOT_EXIST
246	One of the following: • The capture or credit is not voidable because the capture or credit information has already been submitted to your processor. • You requested a void for a type of transaction that cannot be voided. Possible action: No action required.	AUTHORIZATION_CANNOT_BE_VOIDED ORDER_CANNOT_BE_VOIDED