

# Klarna

Simple Order API



**Developer Guide**



**cybersource**  
A Visa Solution

© 2023. 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 Unpublished rights reserved under the copyright laws of the United States.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 and Cybersource Decision Manager 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: 23.01

# Contents

<b>Recent Revisions to This Document.....</b>	<b>5</b>
<b>About This Guide.....</b>	<b>6</b>
<b>Integrating with Klarna.....</b>	<b>7</b>
Merchant Accounts.....	9
Klarna Workflow.....	10
Calculating the Grand Total.....	13
<b>Create a Session.....</b>	<b>17</b>
Line Items.....	19
Create a HOP Session.....	20
Create an Inline Session.....	24
XML Example: Creating a Session.....	27
Name-Value Pair Example: Creating a Session.....	30
<b>Update a Session.....</b>	<b>33</b>
Required Fields for Updating a Session.....	35
Optional Field for Updating a Session.....	38
Coupon Fields.....	39
XML Example: Updating a Session.....	40
Name-Value Pair Example: Updating a Session.....	42
<b>Authorizations.....</b>	<b>43</b>
Required Fields for an Authorization.....	45
XML Example: Authorization.....	47
Name-Value Pair Example: Authorization.....	49
<b>Follow-on Authorizations.....</b>	<b>50</b>
Required Fields for a Follow-on Authorization.....	51
XML Example: Follow-on Authorization.....	53
Name-Value Pair Example: Follow-on Authorization.....	55
<b>Check Status.....</b>	<b>56</b>
Required Fields for Check Status.....	57
XML Example: Check Status.....	58
Name-Value Pair Example: Check Status.....	59
<b>Authorization Reversals.....</b>	<b>60</b>
Required Fields for Authorization Reversal.....	61

XML Example: Authorization Reversal.....	62
Name-Value Pair Example: Authorization Reversal.....	63
<b>Captures.....</b>	<b>65</b>
Required Fields for a Capture.....	66
Optional Fields for Captures.....	67
XML Example: Capture.....	68
Name-Value Pair Example: Capture.....	69
<b>Refunds.....</b>	<b>71</b>
Required Fields for a Refund.....	72
Optional Fields for Refunds.....	73
XML Example: Refund.....	74
Name-Value Pair Example: Refund.....	75
<b>Reason Codes and Klarna Response Codes.....</b>	<b>76</b>
<b>Reporting.....</b>	<b>77</b>
<b>Simple Order API Field Map.....</b>	<b>79</b>

# Recent Revisions to This Document

## 23.01

The guide has undergone a major reorganization.

## 22.02

Updated **billTo\_language** field description to include more countries and languages.

## 22.01

New feature: Follow-On Authorization allows updates to already submitted orders. Bill to, ship to, purchase total, itemization, and item break up can be updated before Capture or Reversal.

Added fields **merchantDefinedData5** to Authorization. See [Authorizations \(on page 43\)](#).

Updated the Supported Countries and Currencies table. See [Supported Countries and Currencies \(on page 7\)](#).

## 20.02

Indicated that the **Klarna.Credit.load** function enables the display of the Klarna widget in the workflow. The Klarna SDK documentation refers to the **Klarna.Payments.load** function, which is not valid for Cybersource integration. See [Klarna Workflow \(on page 10\)](#).

## 20.01

Updated response values for **apCaptureReply\_paymentStatus** and **ap\_capture\_status** and **apCaptureReply\_status** and **ap\_capture\_status**. Added new information about test triggers and URL. See [Testing \(on page 8\)](#).

## 19.04

This revision contains only editorial changes and no technical updates.

# About This Guide

This section describes how to use this guide and where to find further information.

## Audience and Purpose

This guide is written for merchants who want to offer Klarna payments services to customers. It describes tasks that a merchant must complete in order to make a payment, request the status of a payment, or refund a payment. It is intended to help the merchant provide a seamless customer payment experience.

## Convention

The following statement 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.

## Customer Support

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

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

# Integrating with Klarna

Klarna is a Buy Now Pay Later (BNPL) payment method that you can offer your customers through Cybersource. With Klarna, you can enable your customers to split their payments into multiple installments. You can display the Klarna payment method to your customers during checkout by either presenting a Klarna widget or redirecting your customers to a Klarna-hosted page.



**Important:** If you integrated with Klarna through Cybersource before September 2023, review this document and reintegrate to the latest fields and values.

## Requirements

You must obtain a Cybersource merchant ID and a Klarna API key for each country in which you process transactions. Contact your Cybersource account manager for further details.

## Supported Countries and Currencies

Contact your account manager for the latest supported countries and currencies information.

For information about the country codes, currency codes, and language codes, see the *ISO Standard Country Codes*, the *ISO Standard Currency Codes*, and the *ISO Standard Language Codes*.

## Klarna Review Process

Before you launch Klarna payments, Klarna reviews your integrations. For more information about Klarna's pre-launch review process, contact your Cybersource account manager.

## Shipping Policies

Always follow the shipping policies for each country as outlined by Klarna to ensure that Klarna assumes liability for fraudulent transactions. For Klarna's shipping policy, see:

<https://www.klarna.com/international/shipping-policies/>

## Disputes and Fraud

Klarna has a standard process for handling risky transactions and disputes between you and your customers. For more information, contact your technical account manager or customer support.

## Testing

For test transactions, send requests to the Cybersource test server using version 1.187 or later:

```
https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor
```

Use the Klarna test triggers when directly connected to Klarna, which consist of email addresses. For more information, see:

```
https://docs.klarna.com/resources/test-environment/
```

# Merchant Accounts

There are two types of Cybersource merchant accounts, Cybersource settlement services account and processor direct contract account.

## Cybersource Settlement Services Account

This merchant account has no direct contract with a payment provider partner. The Cybersource Financial Settlement Partner (FSP) collects funds on your behalf and settles them to your merchant account.



**Important:** Cybersource requests the export compliance service for every transaction using the Cybersource settlement services account. The export compliance service compares customer information to export control lists maintained by government agencies. If a customer's name appears on any government list, the transaction is declined.

To facilitate compliance checks for Cybersource settlement services accounts, you must include these fields in your authorization service requests.

**billTo\_city**

**billTo\_country**

**billTo(firstName**

**billTo(lastName**

**billTo\_street1**

You must include these fields in the live environment to avoid receiving error messages.

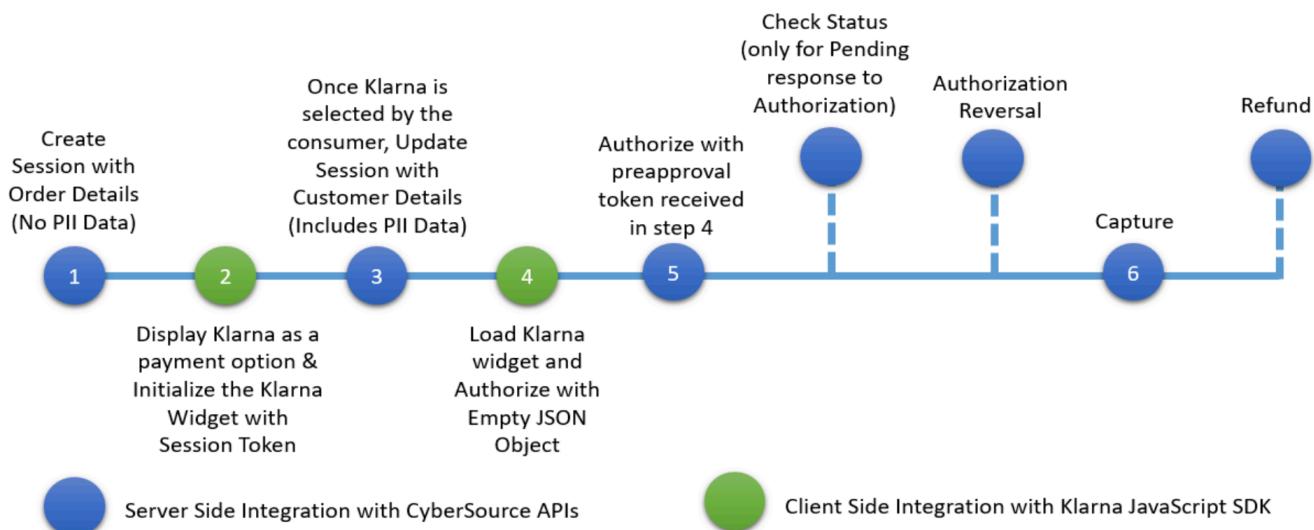
## Processor Direct Contract Account

This merchant account must use the payment provider selected by Cybersource. If you have existing direct contracts, you must inform your sales representative.

# Klarna Workflow

This workflow describes a successful Klarna transaction.

**Figure 1     Klarna Transaction**



1. Add the container for the Klarna widget to the HTML for your checkout page. This is a one-time operation and provides an iFrame for the Klarna widget to be dynamically loaded into when the Klarna widget is initialized.

## Example: Adding the Klarna Container to Your Checkout Page

```
<div id="#klarna_container"></div>
```

2. When the customer displays your checkout page, send a create session request to CyberSource. The sessions service creates a unique customer session and returns a pre-approval token. See [Create a Session \(on page 17\)](#) and [Update a Session \(on page 33\)](#).
3. Present the available payment methods to the customer. When the customer chooses the Klarna payment method on your checkout page, install the Klarna software developer kit (SDK) and initialize it by calling **Klarna.Payments.init**. In the request, set the client token field to the value of the pre-approval token returned by CyberSource.

Initializing the Klarna SDK can take up to 10 seconds. CyberSource recommends that you try to initialize the SDK every three seconds, up to a maximum of three attempts.

For additional information about initializing the Klarna SDK, see:

<https://docs.klarna.com/klarna-payments/in-depth-knowledge/klarna-payments-sdk-reference/>

## Example: Initializing the Klarna SDK

```
Klarna.Payments.init({client_token: '<%=processorToken%>' })
  if (count < 3)
  {
    setTimeout(initializeKlarna.bind(null, count), 3000);
  }
  else
  {
    showError()
  }
```

4. Load the Klarna widget into the Klarna container by calling **Klarna.Payments.load** and specifying the Klarna container.

## Example: Loading the Klarna Widget

```
Klarna.Payments.load({
  container: "#klarna_container",
  ...
})
```

5. Display the Klarna payment options on your checkout page. The `show_form = true` statement dynamically updates the payment options in the Klarna widget.

## Example: Displaying the Klarna Payment Options

```
if (res["show_form"] == true)
{
  logging("Klarna Available Payment Option");

  document.getElementById("auth_button").innerHTML =
  "<br><button type=\"button\" name=\"buy\" onclick=\"authorizeKlarnaOrder();\">Pay</button>"
}

else
{
  logging("Klarna Not Available As A Payment Option");
}
```

6. When the customer chooses one of the Klarna payment options:

- Send an update session request to Cybersource with available customer information. See [Create a Session \(on page 17\)](#) and [Update a Session \(on page 33\)](#).
- Call **Klarna.Payments.authorize** to authorize the order with Klarna. In the call, include an empty JSON object. For additional information about Klarna authorizations, see:

```
https://docs.klarna.com/klarna-payments/api-call-descriptions/authorize-the-purchase/
```

## Example: Authorizing the Order

```
Klarna.Payments.authorize({}, function(res) {  
    var auth_token = res["authorization_token"];  
    var isApproved = res["approved"];  
    var show_form = res["show_form"];  
})
```

7. Klarna validates the customer's information and determines whether to authorize the order. When Klarna authorizes the order, Klarna returns an authorization token.
8. Send an authorization request to Cybersource. Set the pre-approval token field to the value of the authorization token returned by Klarna. See [Authorizations \(on page 43\)](#).
9. When Cybersource approves the authorization, send a capture request to complete the purchase. See [Captures \(on page 65\)](#).

When the authorization response indicates that the purchase is pending, send a check status request every hour until the payment status changes. See [Check Status \(on page 56\)](#).

# Calculating the Grand Total

Most Klarna services require that the grand total amount of a purchase be included in the service request in the **purchaseTotals\_grandTotalAmount** field. The country of the transaction and the use of coupons affect how to calculate the grand total amount.

## US Grand Total with Coupons

To calculate the grand total amount for US transactions with coupons, use this formula:

sum of (unit price x quantity) for all items + item-level tax amount – sum of (**coupon** amount x quantity) for all items – item-level discount amount

This is the same formula with the respective API fields:

sum of (**item\_#\_unitPrice** x **item\_#\_quantity**) for all items + **purchaseTotals\_taxAmount** – sum of coupons (**item\_#\_unitPrice** x **item\_#\_quantity**) – **purchaseTotals\_discountAmount**

## US Grand Total Amount with Coupons

This example shows the proper syntax and calculation of a grand total amount within an API request.

```
<item id="0">
  <unitPrice>100</unitPrice>
  <quantity>1</quantity>
  <totalAmount>100</totalAmount>
</item>
<item id="1">
  <unitPrice>75</unitPrice>
  <quantity>2</quantity>
  <totalAmount>150</totalAmount>
</item>
<item id="2">
  <unitPrice>30</unitPrice>
  <quantity>1</quantity>
  <productCode>coupon</productCode>
  <productName>first-time customer</productName>
  <productSKU>12345</productSKU>
</item>
<purchaseTotals>
  <taxAmount>40</taxAmount>
  <discountAmount>10</discountAmount>
  <grandTotalAmount>250</grandTotalAmount>
</purchaseTotals>
```

## US Grand Total without Coupons

To calculate the grand total amount for US transactions without coupons, use this formula:  
sum of (unit price x quantity) for all items + item-level tax amount – item-level discount amount

This is the same formula with the respective API fields:

sum of (**item\_#\_unitPrice** x **item\_#\_quantity**) for all items + **purchaseTotals\_taxAmount** – **purchaseTotals\_discountAmount**

## Grand Total Amount

This example shows the proper syntax and calculation of a grand total amount within an API request.

```
<item id="0">
  <unitPrice>100</unitPrice>
  <quantity>1</quantity>
  <totalAmount>100</totalAmount>
</item>
<item id="1">
  <unitPrice>75</unitPrice>
  <quantity>2</quantity>
  <totalAmount>150</totalAmount>
</item>
<purchaseTotals>
  <discountAmount>10</discountAmount>
  <taxAmount>40</taxAmount>
  <grandTotalAmount>280</grandTotalAmount>
</purchaseTotals>
```

## Non-US Countries Grand Total with Coupons

To calculate the grand total amount for non-US transactions with coupons, use this formula:  
sum of (unit price x quantity) for all items + sum of (item-level tax amount) for all items – sum of (coupon amount x quantity) for all items – item-level discount amount

This is the same formula with the respective API fields:

sum of (**item\_#\_unitPrice** x **item\_#\_quantity**) for all items + sum of (**item\_#\_taxAmount**) for all items – sum of coupons (**item\_#\_unitPrice** x **item\_#\_quantity**) for all times – **purchaseTotals\_discountAmount**

## Grand Total Amount

This example shows the proper syntax and calculation of a grand total amount within an API request.

```

<item id="0">
  <unitPrice>100</unitPrice>
  <quantity>1</quantity>
  <taxAmount>20</taxAmount>
  <totalAmount>120</totalAmount>
</item>
<item id="1">
  <unitPrice>75</unitPrice>
  <quantity>2</quantity>
  <taxAmount>20</taxAmount>
  <totalAmount>170</totalAmount>
</item>
<item id="2">
  <unitPrice>30</unitPrice>
  <quantity>1</quantity>
  <productCode>coupon</productCode>
  <productName>first-time customer</productName>
  <productSKU>12345</productSKU>
<purchaseTotals>
  <discountAmount>10</discountAmount>
  <grandTotalAmount>290</grandTotalAmount>
</purchaseTotals>

```

## Non-US Countries Grand Total without Coupons

To calculate the grand total amount for non-US transactions without coupons, use this formula:  
 sum of (unit price x quantity) for all items + sum of (item-level tax amount) for all items – item-level discount amount

This is the same formula with the respective API fields:

sum of (**item\_#\_unitPrice** x **item\_#\_quantity**) for all items + sum of (**item\_#\_taxAmount**) for all items – **purchaseTotals\_discountAmount**

## Grand Total Amount

This example shows the proper syntax and calculation of a grand total amount within an API request.

```

<item id="0">
  <unitPrice>100</unitPrice>
  <quantity>1</quantity>
  <totalAmount>120</totalAmount>
</item>
<item id="1">
  <unitPrice>75</unitPrice>

```

```
<quantity>2</quantity>
<taxAmount>20</taxAmount>
<totalAmount>170</totalAmount>
</item>
<purchaseTotals>
  <discountAmount>10</discountAmount>
  <grandTotalAmount>280</grandTotalAmount>
</purchaseTotals>
```

# Create a Session

Create a new session whenever the customer displays your checkout page. The session service responds with a processor token from Klarna. Use this token when initializing the Klarna SDK on your checkout page. Once the customer approves the purchase and consents to sharing their data with Klarna, update the session.

## Payment Flow Mode

When requesting the session service, you must determine how to display the Klarna payment option to your customers. To display Klarna as either an interactive widget or a Klarna-hosted redirect page, set the **apSessionsService\_paymentFlowMode** field to one of these values:

- **HOP**: Redirects the consumer to a Klarna-hosted page. See [Create a HOP Session \(on page 20\)](#).
- **inline**: Presents the Klarna widget into page. See [Create an Inline Session \(on page 24\)](#).

## Payment Methods

When you request the session service, you must determine the Klarna payment method you want to offer your customers. Set the **paymentMethod\_name** field to a value listed in the Payment Method column.

### Payment Methods

Payment Method	Description	Klarna Products
pay_later	Customer pays total bill at a set date.	<ul style="list-style-type: none"><li>• Klarna Invoice</li><li>• Klarna Pay Later</li><li>• Klarna Pay Later by Card (PLBC)</li></ul>
pay_now	Customer pays total bill at the time of checkout.	<ul style="list-style-type: none"><li>• Klarna Direct Bank Transfer</li><li>• Klarna Direct Debit</li></ul>
pay_over_time	Customer pays bill in equal multiple installments.	<ul style="list-style-type: none"><li>• Klarna Account</li><li>• Klarna Financing</li><li>• Klarna Fixed Installment Plans</li><li>• Klarna Interest Free</li></ul>

### Example: Payment Methods in Session Request

```
<paymentMethod_name>pay_later</paymentMethod_name>
```

## Response Status

The session service responds with one of these statuses as the **apSessionReply\_status** field value:

- **COMPLETED**: The customer completed the checkout process.
- **INITIATED**: The session is initiated and the customer may now checkout.

The session service also responds with a reason code as the **apSessionReply\_reasonCode** field value. For more information on reason codes, see the [Reason Codes for the Simple Order API](#).

## Line Items

Klarna uses line items when you send a create session and update session requests. *Line items* are used to include information about the goods that your customers purchase, such as product name, quantity, and price.

Line items are represented as the **item\_#\_** fields, starting with **item\_0\_**, and increasing in numerical order.

These fields are required for each line item that you use:

**item\_#\_productCode**  
**item\_#\_productDescription**  
**item\_#\_productName**  
**item\_#\_quantity**  
**item\_#\_unitPrice**

## Including Line Items in a Service Request

This example shows three valid line items.

```
<item id="0">
  <productCode>123456</productCode>
  <productDescription>Red</productDescription>
  <productName>Shirt</productName>
  <quantity>1</quantity>
  <unitPrice>30.00</unitPrice>
</item>
<item id="1">
  <productCode>456789</productCode>
  <productDescription>Green</productDescription>
  <productName>Pants</productName>
  <quantity>3</quantity>
  <unitPrice>19.99</unitPrice>
</item>
<item id="2">
  <productCode>987654</productCode>
  <productDescription>Blue</productDescription>
  <productName>Dress</productName>
  <quantity>2</quantity>
  <unitPrice>25.50</unitPrice>
</item>
```

# Create a HOP Session

Follow these steps to successfully complete a HOP Session.

1. Send a `POST` request to the <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor> endpoint and include these required fields:



**Important:** Do not send any personally identifiable information (PII) data about the customer in the request to create a session.

## **apPaymentType**

Set to `KLI`.

## **apSessionService\_paymentFlowMode**

Set to `HOP` to display the Klarna widget on your checkout page.

## **apSessionsService\_cancelURL**

Set to the URL the customer is directed to after cancelling the Klarna payment.

## **apSessionsService\_failureURL**

Set to the URL the customer is directed to after the Klarna payment fails.

## **apSessionsService\_paymentMethod\_name**

Set to one of these values to display the Klarna product you are offering:

- `pay_later`: pay later
- `pay_now`: pay now
- `pay_over_time`: pay over time

## **apSessionsService\_run**

Set to `true`.

## **apSessionsService\_sessionsType**

Set to `N`.

## **apSessionsService\_successURL**

Set to the URL the customer is directed to after successfully completing the Klarna payment.

Include `?auth-token={{authorization_token}}` to the end of the URL.

### **Example:**

```
https://www.merchant.com?auth-token={{authorization_token}}
```

**billTo\_country**

**item\_#\_productName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_quantity**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_totalAmount**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_unitPrice**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**merchantID**

**merchantReferenceCode**

**purchaseTotals\_currency**

**purchaseTotals\_grandTotalAmount**

**Non-US Countries**

**item\_#\_taxAmount**

**US**

**billTo\_state**

**purchaseTotals\_taxAmount**

2. Include any of these optional fields in the request:

**apSessionsService\_sessionsType**

**purchaseTotals\_discountAmount**

3. To include a coupon in the request, include these fields:

**item\_#\_productCode**

Set to [coupon](#).

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_productName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **item\_#\_productSKU**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **item\_#\_quantity**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **item\_#\_unitPrice**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

4. If the customer changes their item selection after you sent the Create Session request, send an Update Session request. See [Update a Session \(on page 33\)](#).
5. Store the value in the **processorToken** field from the session response. You must include this value in the authorization request.

#### **Example: Returned Token**

```
<processorToken>d9b4496b-f4b1-5ad5-bbe0-7f666682126</processorToken>
```

6. Redirect the customer to the returned URL in the **merchantURL** field.

#### **Example: Merchant URL from Session Response**

```
<merchantURL>https://pay.playground.klarna.com/na/hpp/payments/1vmXvue</merchantURL>
```

7. When the customer completes the checkout using their Klarna credentials, Klarna redirects the customer to the URL you specified in the **apSessionsService\_successURL** field. This URL contains a unique token.

#### **Example: Redirect Success URL**

```
https://www.merchant.com?auth-token=d9b4496b-f4b1-5ad5-bbe0-7f666682126
```

- When you send the authorization request, set the **apAuthService\_preapprovalToken** field to the unique token Klarna returned in the success URL.

#### Example: Pre-approval Token in Authorization Request

```
<preapprovalToken>d9b4496b-f4b1-5ad5-bbe0-7f666682126</preapprovalTok  
en>
```

#### Related information

[API Field Reference for the Simple Order API](#)

# Create an Inline Session

Follow these steps to successfully complete an inline Session.

1. Send a `POST` request to the <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor> endpoint and include these required fields:



**Important:** Do not send any personally identifiable information (PII) data about the customer in the request to create a session.

## **apPaymentType**

Set to `KLI`.

## **apSessionService\_paymentFlowMode**

Set to `inline` to display the Klarna widget on your checkout page.

## **apSessionsService\_cancelURL**

Set to the URL the customer is directed to after cancelling the Klarna payment.

## **apSessionsService\_failureURL**

Set to the URL the customer is directed to after the Klarna payment fails.

## **apSessionsService\_paymentMethod\_name**

Set to one of these values to display the Klarna product you are offering:

- `pay_later`: pay later
- `pay_now`: pay now
- `pay_over_time`: pay over time

## **apSessionsService\_run**

Set to `true`.

## **apSessionsService\_sessionsType**

Set to `N`.

## **apSessionsService\_successURL**

Set to the URL the customer is directed to after successfully completing the Klarna payment.

Include `/Klarna/Reply.jsp` to the end of the URL.

### **Example:**

```
https://www.merchant.com/Klarna/Reply.jsp
```

**billTo\_country**

**item\_#\_productName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_quantity**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_totalAmount**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_unitPrice**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**merchantID**

**merchantReferenceCode**

**purchaseTotals\_currency**

**purchaseTotals\_grandTotalAmount**

**Non-US Countries**

**item\_#\_taxAmount**

**US**

**billTo\_state**

**purchaseTotals\_taxAmount**

2. Include any of these optional fields in the request:

**apSessionsService\_sessionsType**

**purchaseTotals\_discountAmount**

3. To include a coupon in the request, include these fields:

**item\_#\_productCode**

Set to [coupon](#).

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_productName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **item\_#\_productSKU**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **item\_#\_quantity**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **item\_#\_unitPrice**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

4. If the customer changes their item selection after you sent the Create Session request, send an Update Session request. See [Update a Session \(on page 33\)](#).

5. Set the Session Token ID in the Klarna widget to the value in the **processorToken** returned in the Session response.

#### **Example: Session Token ID in Klarna Widget**

```
IDeyJhbGciOiJSUzI1NiIsImtpZCI6IjgyMzA1ZWJjLWI4MTEtMzYzNy... .
```

6. When the customer completes the payment using the Klarna widget, the Klarna widget displays a unique token. Store the token for the authorization request.

#### **Example: Returned Authorization Token**

```
Authorizaiton Token: a6b2229c-665b-5660-a5fe-063bef3d9a1e
```

7. When you begin to authorize the payment, set the **apAuthService\_preapprovalToken** field in the authorization request to the token Klarna displayed in the Klarna widget.

#### **Example: Pre-approval Token in Authorization Request**

```
<preapprovalToken>a6b2229c-665b-5660-a5fe-063bef3d9a1e</preapprovalTok  
en>
```

## **Related information**

[API Field Reference for the Simple Order API](#)

# XML Example: Creating a Session

This example shows a successful create session request.

## HOP Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantID>merch_123</merchantID>
<merchantReferenceCode>ref_123456</merchantReferenceCode>
<billTo>
<firstName>John</firstName>
<lastName>Smith</lastName>
<street1>123 Happy St</street1>
<city>Sunnyville</city>
<state>VA</state>
<postalCode>12345</postalCode>
<country>US</country>
<email>john@email.com</email>
</billTo>
<item>
<unitPrice>150.00</unitPrice>
<quantity>1</quantity>
<productCode>A4890B5023</productCode>
<productName>Skirt on the sky</productName>
<productSKU>skirtovsky$bluegreen</productSKU>
<taxAmount>4.50</taxAmount>
<totalAmount>154.50</totalAmount>
<productDescription>Amnesiac Shirt</productDescription>
</item>
<purchaseTotals>
<currency>USD</currency>
<grandTotalAmount>154.50</grandTotalAmount>
</purchaseTotals>
<apPaymentType>kli</apPaymentType>
<apSessionsService run="true">
<cancelURL>https://www.merchant.com</cancelURL>
<successURL>https://www.merchant.com?auth-token={{authorization_token}}</successURL>
<failureURL>https://www.merchant.com</failureURL>
<sessionsType>N</sessionsType>
<paymentMethod_name>pay_now</paymentMethod_name>
<paymentFlowMode>HOP</paymentFlowMode>
</apSessionsService>
</requestMessage>
```

## HOP Response

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantReferenceCode>Postman-1695755654</merchantReferenceCode>
<requestID>6957556550046839703010</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>AxjnrwSTeSt+U42aL4Xi/6IZYjWmdqbZrNGMSeojmjL6m3SB3i0ZiwyASzejFlof+k3krfl0
Nmi+F4gA7yG1</requestToken>
<purchaseTotals>
<currency>USD</currency>
</purchaseTotals>
<apSessionsReply>
<reasonCode>100</reasonCode>
<responseCode>00000</responseCode>
<merchantURL>https://pay.playground.klarna.com/na/hpp/payments/1vmXvue</merchantURL>
<processorToken>eyJhbGciOiJSUzI1NiIsI_TEST_TOKEN_I4MTEtMzYzNy1hYTRjLTY2ZW</processorTok
en>
<amount>154.50</amount>
<reconciliationID>XFZ3ZMYV41D0</reconciliationID>
<status>COMPLETED</status>
<dateTime>2023-09-26T19:14:16Z</dateTime>
</apSessionsReply>
</replyMessage>
```

## Inline Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantID>merch_test</merchantID>
<merchantReferenceCode>ref-123456</merchantReferenceCode>
<billTo>
<firstName>John</firstName>
<lastName>Smith</lastName>
<street1>123 Happy St</street1>
<city>Sunnyville</city>
<state>VA</state>
<postalCode>12345</postalCode>
<country>US</country>
<email>john@email.com</email>
</billTo>
<item>
<unitPrice>9.00</unitPrice>
<quantity>1</quantity>
<productCode>A4890B5023</productCode>
<productName>Skirt on the sky</productName>
<productSKU>skirtonsky$bluegreen</productSKU>
<taxAmount>1.00</taxAmount>
```

```

<totalAmount>10.00</totalAmount>
<productDescription>Amnesiac Shirt</productDescription>
</item>
<purchaseTotals>
  <currency>USD</currency>
  <grandTotalAmount>10.00</grandTotalAmount>
</purchaseTotals>
<apPaymentType>KLI</apPaymentType>
<apSessionsService run="true">
  <cancelURL>https://www.merchant.com/Klarna/Reply.jsp</cancelURL>
  <successURL>https://www.merchant.com/Klarna/Reply.jsp</successURL>
  <failureURL>https://www.merchant.com/Klarna/Reply.jsp</failureURL>
  <sessionsType>N</sessionsType>
  <paymentMethod_name>pay_now</paymentMethod_name>
  <paymentFlowMode>inline</paymentFlowMode>
</apSessionsService>
</requestMessage>

```

## Inline Response

```

<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.213">
  <merchantReferenceCode>Postman-1695860125</merchantReferenceCode>
  <requestID>6958601255246857703007</requestID>
  <decision>ACCEPT</decision>
  <reasonCode>100</reasonCode>
  <requestToken>AxjnrwSTeTn93WXAQw5f/6IZYjWmdmbZq0orKCojmjrTtjSBX0GY0MmkmXoxZaH/pN50f3dZcBD
D18AA9DN1</requestToken>
  <purchaseTotals>
    <currency>USD</currency>
  </purchaseTotals>
  <apSessionsReply>
    <reasonCode>100</reasonCode>
    <responseCode>00000</responseCode>
    <processorToken>eyJhbGciOiJSUzI1NiIsIm_TEST_TOKEN_1ZWJjlWI4MTEtMzYzNy1hYTRjLTY2ZWNhM</pr
ocessorToken>
    <amount>10.00</amount>
    <reconciliationID>XFZ3YMYURE2A</reconciliationID>
    <status>COMPLETED</status>
    <dateTime>2023-09-28T00:15:26Z</dateTime>
  </apSessionsReply>
</replyMessage>

```

# Name-Value Pair Example: Creating a Session

This example shows a successful create session request.

## HOP Request

```
merchantID=altpay_test
merchantReferenceCode=TC_sessions-1
billTo_state=CA
billTo_postalCode=41460
billTo_country=DE
item_unitPrice=20.00
item_quantity=15
item_productCode=A4890B5023
item_productName=Skirt on the sky
item_productSKU=skirtonsky$bluegreen
item_taxAmount=99.00
item_totalAmount=399.00
item_productDescription=female skirt in blue and green
purchaseTotals_currency=EUR
item_grandTotalAmount=399.0
apPaymentType=KLI
apSessionsService_run="true"
cancelURL=http://test.com/Klarna/Reply.jsp</cancelURL>
successURL=http://test.com/Klarna/Reply.jsp?auth-token={{authorization_token}}
failureURL=http://test.com/Klarna/Reply.jsp</failureURL>
paymentMethod_name=pay_now
paymentFlowMode=HOP
```

## HOP Response

```
merchantReferenceCode=TC_sessions-1
requestID=6316194410506159904010
decision=ACCEPT
reasonCode=100
requestToken=AxjnrwSTVmbOR7kfc9UK/6IZYjWmdmpJtWqrGMojilkyCH0iE3uCdMMmkmXoxX6rUJNWZs5HuR9z1
QoAxD1+
purchaseTotals_currency=EUR
apSessionsReply_reasonCode=100
apSessionsReply_responseCode=00000
merchantURL=https://pay.playground.klarna.com/eu/hpp/payments/9r3hZn1
processorToken=eyJhbGciOiJSUzI1_TEST_TOKENzA1ZWJjLWI4MTEtMzYzNy1hYTR
apSessionReply_amount=399.00
```

```
apSessionsReply_reconciliationID=XFZ3YTIZZU1F  
apSessionsReply_status=COMPLETED  
apSessionsReply_dateTime=2021-09-14T11:37:21Z
```

## Inline Request

```
merchantID=altpay_test  
merchantReferenceCode=TC_sessions-1  
billTo_state=CA  
billTo_postalCode=41460  
billTo_country=DE  
item_unitPrice=20.00  
item_quantity=15  
item_productCode=A4890B5023  
item_productName=Skirt on the sky  
item_productSKU=skirtonsky$bluegreen  
item_taxAmount=99.00  
item_totalAmount=399.00  
productDescription=female skirt in blue and green  
purchaseTotals_currency=EUR  
purchaseTotals_grandTotalAmount=399.0  
apPaymentType=KLI  
apSessionsService_run=true  
sessionsType=N  
paymentMethod_name=pay_now  
paymentFlowMode=inline
```

## Inline Response

```
merchantReferenceCode=TC_sessions-1  
requestID=6316179295726205204008  
decision=ACCEPT  
reasonCode=100  
requestToken=AxjnrwSTVmaYl0ktZPYo/6IZYjWmdmpJtWqrCWojilkVuT0iE3uCdMMmkmXoxX6rUJNWZpiU6S1k9  
igAxAuP  
purchaseTotals_currency=EUR  
apSessionsReply_reasonCode=100  
apSessionsReply_responseCode=00000  
apSessionsReply_processorToken=eyJhbGciOi_TEST_TOKEN_6IjgyMzA1ZWJjLWI4MTEtM  
apSessionsReply_amount=399.00
```

apSessionsReply\_reconciliationID=XFZ3YTIZZU0K  
apSessionsReply\_status=COMPLETED  
apSessionsReply\_dateTime=2021-09-14T11:12:10Z

# Update a Session

You can update a session for 48 hours after creating it. You can update item-level field details and amounts in the request to update the session.

The session update service enables you to update items in the cart, with or without sending customer billing details. However, when you send customer billing details in the session update request, send as many billing details as possible for the best customer checkout experience.

Only update a session after the customer consents to sharing their data with Klarna.

**!** **Important:** PII data about a customer should be sent only after the customer chooses the Klarna payment option on your checkout page and consents to sharing their data with Klarna.

## Payment Methods

When you request the session service, you must determine the Klarna payment method you want to offer your customers. Set the **paymentMethod\_name** field to a value listed in the Payment Method column.

### Payment Methods

Payment Method	Description	Klarna Products
pay_later	Customer pays total bill at a set date.	<ul style="list-style-type: none"><li>• Klarna Invoice</li><li>• Klarna Pay Later</li><li>• Klarna Pay Later by Card (PLBC)</li></ul>
pay_now	Customer pays total bill at the time of checkout.	<ul style="list-style-type: none"><li>• Klarna Direct Bank Transfer</li><li>• Klarna Direct Debit</li></ul>
pay_over_time	Customer pays bill in equal multiple installments.	<ul style="list-style-type: none"><li>• Klarna Account</li><li>• Klarna Financing</li><li>• Klarna Fixed Installment Plans</li><li>• Klarna Interest Free</li></ul>

### Example: Payment Methods in Session Request

```
<paymentMethod_name>pay_later</paymentMethod_name>
```

## Response Status

The session service responds with one of these statuses as the **apSessionReply\_status** field value:

- **COMPLETED**: The customer completed the checkout process.
- **INITIATED**: The session is initiated and the customer may now checkout.

The session service also responds with a reason code as the **apSessionReply\_reasonCode** field value. For more information on reason codes, see the [Reason Codes for the Simple Order API](#).

## Endpoint

Set the **apSessionsService\_run** field to `true` and send the request to one of these endpoints.

- **Production**: <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>
- **Test**: <https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

# Required Fields for Updating a Session

Include these required fields to update a session.

## **apPaymentType**

Set to [KLT](#).

## **apSessionsService\_cancelURL**

Set to the URL the customer is directed to after cancelling the Klarna payment.

## **apSessionsService\_failureURL**

Set to the URL the customer is directed to after the Klarna payment fails.

## **apSessionsService\_requestID**

## **apSessionsService\_run**

Set to [true](#).

## **apSessionsService\_sessionsType**

Set to [U](#).

## **apSessionsService\_successURL**

Set to the URL the customer is directed to after successfully completing the Klarna payment.

## **billTo\_city**

## **billTo\_country**

## **billTo\_district**

Set to the same value as the **billTo\_state** field if the field is present.

## **billTo\_email**

## **billTo(firstName)**

## **billTo\_language**

For possible values, see the country and language code column in the [Supported Countries and Currencies \(on page 7\)](#) table.

## **billTo\_lastName**

## **billTo\_postalCode**

## **billTo\_state**

Set to the same value as the **billTo\_district** field.

## **billTo\_street1**

## **billTo\_street2**

## **item\_{#}\_productName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **item\_#\_quantity**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **item\_#\_totalAmount**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **item\_#\_unitPrice**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

#### **merchantID**

#### **merchantReferenceCode**

#### **purchaseTotals\_currency**

#### **purchaseTotals\_grandTotalAmount**

#### **shipTo\_city**

#### **shipTo\_country**

#### **shipTo\_district**

#### **shipTo\_email**

#### **shipTo(firstName)**

#### **shipTo(lastName)**

#### **shipTo(postalCode)**

#### **shipTo(state)**

#### **shipTo(street1)**

#### **shipTo(street2)**

## **Country-Specific Fields**

Include the country-specific field in addition to the required fields for a session.

#### **Non-US Countries**

#### **item\_#\_taxAmount**

#### **US**

**billTo\_state**

**purchaseTotals\_taxAmount**

**Related information**

[API Field Reference for the Simple Order API](#)

# Optional Field for Updating a Session

You can include this optional field to update a session.

**`purchaseTotals_discountAmount`**

## Related information

[API Field Reference for the Simple Order API](#)

# Coupon Fields

To include a coupon in your request, include these fields.

## **item\_#\_productCode**

Set to [coupon](#).

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_productName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_productSKU**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_quantity**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_unitPrice**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **Related information**

[API Field Reference for the Simple Order API](#)

# XML Example: Updating a Session

These examples show a successful updated session requests.

## Request

```
<requestMessage
  xmlns="urn:schemas-cybersource-com:transaction-data-1.213">
  <merchantID>mid123</merchantID>
  <merchantReferenceCode>ref1234</merchantReferenceCode>
  <billTo>
    <firstName>Anna</firstName>
    <lastName>Schmidt</lastName>
    <street1>Leopoldstrasse 4</street1>
    <street2>Apt 2</street2>
    <city>Lichtenberg</city>
    <district>Berlin</district>
    <state>Berlin</state>
    <postalCode>10318</postalCode>
    <country>DE</country>
    <phoneNumber>5551234567</phoneNumber>
    <email>schmidt@example.com</email>
    <dateOfBirth>19820101</dateOfBirth>
    <language>DE-DE</language>
  </billTo>
  <shipTo>
    <firstName>Anna</firstName>
    <lastName>Schmidt</lastName>
    <street1>Leopoldstrasse 4</street1>
    <street2>Apt 2</street2>
    <city>Lichtenberg</city>
    <state>Berlin</state>
    <district>Berlin</district>
    <postalCode>10318</postalCode>
    <country>DE</country>
    <phoneNumber>5551234567</phoneNumber>
    <email>schmidt@example.com</email>
  </shipTo>
  <item id="0">
    <unitPrice>19.99</unitPrice>
    <quantity>1</quantity>
    <productName>Green Widget</productName>
    <taxAmount>1.00</taxAmount>
    <totalAmount>20.99</totalAmount>
  </item>
  <item id="1">
    <unitPrice>10.00</unitPrice>
```

```
<quantity>2</quantity>
<productName>Blue Widget</productName>
<taxAmount>2.00</taxAmount>
<totalAmount>22.00</totalAmount>
</item>
<item id="2">
<unitPrice>5.00</unitPrice>
<quantity>1</quantity>
<productName>shipping</productName>
<totalAmount>5.00</totalAmount>
</item>
<purchaseTotals>
<currency>EUR</currency>
<discountAmount>3.00</discountAmount>
<grandTotalAmount>44.99</grandTotalAmount>
</purchaseTotals>
<apPaymentType>KLI</apPaymentType>
<apSessionsService run="true">
<sessionsType>U</sessionsType>
<sessionsRequestID>4848446567036715804007</sessionsRequestID>
</apSessionsService>
</requestMessage>
```

## Response

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantReferenceCode>ref1234</merchantReferenceCode>
<requestID>4848446567036715804007</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<purchaseTotals>
<currency>EUR</currency>
</purchaseTotals>
<apSessionsReply>
<reasonCode>100</reasonCode>
<responseCode>00000</responseCode>
</apSessionsReply>
</replyMessage>
```

# Name-Value Pair Example: Updating a Session

These examples show a successful updated session requests.

## Request

```
apSessionsService_run=true
apSessionsService_cancelURL=http://test.com/Klarna/Reply.jsp
apSessionsService_successURL=http://test.com/Klarna/Reply.jsp
apSessionsService_failureURL=http://test.com/Klarna/Reply.jsp
apSessionsService_sessionsType=U
apSessionsService_requestID=4848446567036715804007
merchantID=mid123
merchantReferenceCode=ref1234
billTo(firstName=Anna
billTo(lastName=Schmidt
billTo(street1=Leopoldstrasse 4
billTo(street2=Apt 2
billTo(city=Lichtenberg
billTo(district=Berlin
billTo(state=Berlin
billTo(postalCode=10318
billTo(country=DE
billTo(phoneNumber=5551234567
billTo(email=schmidt@example.com
billTo(dateOfBirth=19820101
billTo(language=DE-DE
shipTo(firstName=Anna
shipTo(lastName=Schmidt
shipTo(street1=Leopoldstrasse 4
shipTo(street2=Apt 2
```

## Response

```
merchantReferenceCode=ref1234
requestID=4848446567036715804007
decision=ACCEPT
reasonCode=100

purchaseTotals_currency=EUR
apSessionsReply_reasonCode=100
apSessionsReply_responseCode=00000
```

# Authorizations

The authorization service responds with a Klarna URL to which you direct the customer after the transaction is complete. The Klarna URL is returned in the **apAuthReply\_merchantURL** field. You can capture an authorization for up to 28 days after a payment is authorized.

## Merchant-Defined Data

Merchant-defined data is included in the **merchantDefinedData\_mddField\_5** field.

**!** **Important:** This field has replaced the **merchantDefinedData\_field5** field. If you submit a request with the **merchantDefinedData\_field5** field, Cybersource automatically replaces the field with **merchantDefinedData\_mddField\_5**.

**!** **Warning:** You are prohibited from including personally identifying information (PII) in the **merchantDefinedData** fields for the purposes of capturing, obtaining, and/or transmitting any PII. PII includes, but is not limited to, address, credit card number, social security number, driver's license, state-issued identification number, passport number, and card verification number (CVV, CVC2, CVV2, CID, CVN). In the event Cybersource discovers you are capturing and/or transmitting PII in the **merchantDefinedData** fields, whether or not intentionally, Cybersource immediately suspends your merchant account, which results in a rejection of any and all transaction requests submitted by the merchant account after the point of suspension.

## Response Status

The authorization service responds with one of these statuses as the **apAuthReply\_status** field value:

- **AUTHORIZED:** The payment is successfully authorized.
- **FAILED:** The authorization request failed.
- **PENDING:** The authorization request is accepted but is not authorized. Request the check status service to retrieve status updates. For more information, see [Check Status \(on page 56\)](#).

The authorization service also responds with a reason code as the **apAuthReply\_reasonCode** field value. For more information on reason codes, see the [Reason Codes for the Simple Order API](#).

## Endpoint

Set the **apAuthService\_run** field to `true` and send the request to one of these endpoints.

- **Production:** <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>
- **Test:** <https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

# Required Fields for an Authorization

Include these required fields to process an authorization.

## **apAuthService\_preapprovalToken**

Set to the token generated by Klarna.

## **apAuthService\_run**

Set to `true`.

## **apPaymentType**

Set to `KLI`.

## **billTo\_city**

## **billTo\_country**

Set to the same value used in the sessions request.

## **billTo\_email**

## **billTo(firstName)**

## **billTo(lastName)**

## **billTo(street1)**

## **merchantDefinedData\_mddField\_5**

Set to the same value used in the create session request. Possible values:

- `HOP`: display the Klarna widget on your checkout page.
- `inline`: redirect the customer to the Klarna-hosted page.

## **merchantID**

## **merchantReferenceCode**

## **purchaseTotals\_currency**

## **purchaseTotals\_grandTotalAmount**

Set to the same value as the **purchaseTotals\_grandTotalAmount** field in the latest sessions update request.

## Country-Specific Fields

Include the country-specific field in addition to the required fields for a session.

### **Non-US Countries**

#### **item#\_taxAmount**

**US**

**billTo\_state**

**purchaseTotals\_taxAmount**

## Follow-on Authorizations

Include this field, in addition to the required fields, to request a follow-on authorization.

**linkToRequest**

Set to the **requestID** from the initial authorization response.

### Related information

[API Field Reference for the Simple Order API](#)

# XML Example: Authorization

This example shows a successful authorization.

## Authorization Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantID>test_merch</merchantID>
<merchantReferenceCode>ref_123456</merchantReferenceCode>
<billTo>
<firstName>John</firstName>
<lastName>Smith</lastName>
<street1>Happy St</street1>
<street2>123</street2>
<city>Austin</city>
<district>TX</district>
<state>TX</state>
<postalCode>78757</postalCode>
<country>US</country>
<email>null@cybersource.com</email>
</billTo>
<purchaseTotals>
<currency>USD</currency>
<grandTotalAmount>154.50</grandTotalAmount>
</purchaseTotals>
<merchantDefinedData>
<field5>inline</field5>
</merchantDefinedData>
<apPaymentType>KLI</apPaymentType>
<apAuthService run="true">
<preapprovalToken>d9b4496b-f4b1-5ad5-bbe0-7f6666821260</preapprovalToken>
</apAuthService>
</requestMessage>
```

## Authorization Response

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantReferenceCode>Postman-1695757061</merchantReferenceCode>
<requestID>6957570615206883603012</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>AxjnrwSTeSuwS73eT7JE/6IZYjWmdqbZrNGpuojmjMMSTSBB3i0ZiwyASZejFlof+k3krsEu
93k+yRAABS1G</requestToken>
<purchaseTotals>
<currency>USD</currency>
```

```
</purchaseTotals>
<exportReply>
  <reasonCode>100</reasonCode>
  <ipCountryConfidence>-1</ipCountryConfidence>
</exportReply>
<apReply>
  <productID>paylaterbycard</productID>
</apReply>
<apAuthReply>
  <reasonCode>100</reasonCode>
  <status>AUTHORIZED</status>
  <processorResponse>00003</processorResponse>
  <amount>154.50</amount>
  <dateTime>2023-09-26T19:37:43Z</dateTime>
  <paymentStatus>authorized</paymentStatus>
  <responseCode>00003</responseCode>
  <reconciliationID>XFZ3ZMYV41J7</reconciliationID>
  <processorTransactionID>a3367116-69e7-4ddf-b1fe-645413b6c63d</processorTransactionID>
</apAuthReply>
</replyMessage>
```

# Name-Value Pair Example: Authorization

This example shows a successful authorization.

## Authorization Request

```
apAuthService_run=true
apAuthService_preapprovalToken=ca322c34-6f2d-5421-b29f-0d10ebadc3bb
apPaymentType=KLI
billTo(firstName=Anna
billTo.lastName=Schmidt
billTo.street1=Leopoldstrasse 4
billTo.city=Lichtenberg
billTo.state=Berlin
billTo.postalCode=10318
billTo.country=DE
billTo_email=schmidt@example.com
purchaseTotals_currency=EUR
purchaseTotals_grandTotalAmount=19.99
merchantID=mid123
merchantReferenceCode=ref1234
```

## Authorization Response

```
merchantReferenceCode=ref1234
requestID=4848446567036715804007
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSTBtW2cINmCVnohjaZCoMrLNy1k07KXFKEb9JpA6cQ1QY9
purchaseTotals_currency=EUR
apAuthReply_reasonCode=100
apAuthReply_status=AUTHORIZED
apAuthReply_processorResponse=00003
apAuthReply_amount=19.99
apAuthReply_paymentStatus=authorized
apAuthReply_merchantURL=https://
credit-na.playground.klarna.com/v1/sessions/9896d517-14bd-4f32-bc4a-184924632747/redirect
apAuthReply_reconciliationID=6LBP2Y395ISZ
apAuthReply_processorTransactionID=1a2a3706-cc17-4fd9-a9d6-b437fc064d8f
```

# Follow-on Authorizations

Already submitted authorizations can be updated as follow-on authorizations before the capture or reversal. Follow-on authorizations allow you to update the billing, shipping, itemization, and item break up information of already submitted orders. Request a new authorization with the **linkToRequest** field to link the follow-on authorization to the initial authorization. Set the **linkToRequest** field to the **requestID** value from the initial authorization response.

## Response Status

The authorization service responds with one of these statuses as the **apAuthReply\_status** field value:

- **AUTHORIZED**: The payment is successfully authorized.
- **FAILED**: The authorization request failed.
- **PENDING**: The authorization request is accepted but is not authorized. Request the check status service to retrieve status updates. For more information, see [Check Status \(on page 56\)](#).

The authorization service also responds with a reason code as the **apAuthReply\_reasonCode** field value. For more information on reason codes, see the [Reason Codes for the Simple Order API](#).

## Endpoint

Set the **apAuthService\_run** field to `true` and send the request to one of these endpoints.

- **Production**: <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>
- **Test**: <https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

# Required Fields for a Follow-on Authorization

Include these required fields to process a follow-on authorization.

## **apAuthService\_preapprovalToken**

Set to the token generated by Klarna.

## **apAuthService\_run**

Set to `true`.

## **apPaymentType**

Set to `KLI`.

## **billTo\_city**

## **billTo\_country**

Set to the same value used in the sessions request.

## **billTo\_email**

## **billTo(firstName)**

## **billTo(lastName)**

## **billTo(street1)**

## **linkToRequest**

Set to the **requestID** from the initial authorization response.

## **merchantDefinedData\_mddField\_5**

Set to the same value used in the create session request. Possible values:

- `HOP`: display the Klarna widget on your checkout page
- `inline`: redirect the customer to the Klarna-hosted page

## **merchantID**

## **merchantReferenceCode**

## **purchaseTotals\_currency**

## **purchaseTotals\_grandTotalAmount**

Set to the same value as the **purchaseTotals\_grandTotalAmount** field in the latest sessions update request.

## Country-Specific Fields

Include the country-specific field in addition to the required fields for a session.

## Non-US Countries

**item#\_taxAmount**

## US

**billTo\_state**

**purchaseTotals\_taxAmount**

## Related information

[API Field Reference for the Simple Order API](#)

# XML Example: Follow-on Authorization

This example shows a successful follow-on authorization request.

## Follow-on Authorization Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantID>test_merch</merchantID>
<merchantReferenceCode>ref-123456</merchantReferenceCode>
<billTo>
<firstName>Jane</firstName>
<lastName>Smith</lastName>
<street1>Happy St</street1>
<street2>456</street2>
<city>Austin</city>
<district>TX</district>
<state>TX</state>
<postalCode>78797</postalCode>
<country>US</country>
<email>null@cybersource.com</email>
</billTo>
<purchaseTotals>
<currency>USD</currency>
<grandTotalAmount>10.00</grandTotalAmount>
</purchaseTotals>
<merchantDefinedData>
<field5>inline</field5>
</merchantDefinedData>
<linkToRequest>6958684668456844304007</linkToRequest>
<apPaymentType>KLI</apPaymentType>
<apAuthService run="true">
<preapprovalToken>9cb0b304-c901-5a58-afd5-06c87d7bb217</preapprovalToken>
</apAuthService>
</requestMessage>
```

## Follow-on Authorization Response

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantReferenceCode>Postman-1695868808</merchantReferenceCode>
<requestID>6958688084846249104011</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>AxjnrwSTeTsyWHVD+dKL/6IZYjWmjCbZrSIzSWojmjst4nSBX0GY0MmkmXoxZaH/pN50zJYdUP5
0osAA8wHb</requestToken>
<purchaseTotals>
```

```
<currency>USD</currency>
</purchaseTotals>
<exportReply>
  <reasonCode>100</reasonCode>
  <ipCountryConfidence>-1</ipCountryConfidence>
</exportReply>
<apReply>
  <productID>paylaterbycard</productID>
</apReply>
<apAuthReply>
  <reasonCode>100</reasonCode>
  <status>AUTHORIZED</status>
  <processorResponse>00003</processorResponse>
  <amount>10.00</amount>
  <dateTime>2023-09-28T02:40:10Z</dateTime>
  <paymentStatus>authorized</paymentStatus>
  <responseCode>00003</responseCode>
  <merchantURL>https://js.playground.klarna.com/na/kp/v1/sessions/f2bebdbb-b46c-551e-a8a4-048a4a185c10/redirect</merchantURL>
  <reconciliationID>XFZ40MYVHF4K</reconciliationID>
  <processorTransactionID>fc2b177e-f70a-4ff6-8847-3ab7b8db6d9c</processorTransactionID>
</apAuthReply>
</replyMessage>
```

# Name-Value Pair Example: Follow-on Authorization

This example shows a successful follow-on authorization request.

## Follow-on Authorization Request

```
merchantReferenceCode=TC_sessions-1
billTo_state=CA
billTo_country=DE
purchaseTotals_currency=EUR
purchaseTotals_grandTotalAmount=399.0
merchantDefinedData_field5=inline
linkToRequest=6316148341646188004010
apPaymentType=KLI
apAuthService_run=true
apAuthService_preapprovalToken=68b47976-6411-3ff3-93f8-673474b785e1
```

## Follow-on Authorization Response

```
merchantReferenceCode=TC_sessions-1
requestID=6316168657716339704012
decision=ACCEPT
reasonCode=100
requestToken=AxjnrwSTVmZyybJ+PoDM/6IZYjWmjCpKm2ZLSyojilj6P70iE3uCdMMmkmXoxX6rUJNWZnLJsN4+g
MwALAIi
purchaseTotals_currency=EUR
apAuthReply_reasonCode=100
apAuthReply_status=AUTHORIZED
apAuthReply_processorResponse=00003
apAuthReply_amount=399.00
apAuthReply_dateTime=2021-09-14T10:54:26Z
apAuthReply_paymentStatus=authorized
apAuthReply_responseCode=00003
apAuthReply_reconciliationID=XFZ40TJMYI4Y
apAuthReply_processorTransactionID=19bff185-a92a-346a-a42a-d79995302a0a
```

# Check Status

Request the check status service when the authorization response status is `PENDING`. A pending status occurs when Klarna reviews an authorization. Cybersource recommends that you request the check status service hourly until the payment status changes.

## Response Status

The check status service responds with one of these statuses in the `apCheckStatusReply_status` field value:

- `ABANDONED`: The customer did not complete the payment using the redirect URL.
- `AUTHORIZED`: The customer's payment is authorized.
- `AUTH-REVERSED`: The authorization is successfully reversed.
- `COMPLETED`: The customer completed the payment transaction.
- `FAILED`: The service request failed. A failed request can be due to either Klarna rejecting the transaction or a technical error.
- `PENDING`: The service request is accepted but is not completed. Request the check status service to retrieve status updates.
- `SETTLED`: The capture request is settled for the requested amount.

## Endpoint

Set the `apCheckStatusService_run` field to `true` and send the request to one of these endpoints.

- **Production:** <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>
- **Test:** <https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

# Required Fields for Check Status

Include these required fields to check the status of a transaction.

## **apCheckStatusService\_checkStatusRequestID**

Set to the request ID included in the service response you are checking.

## **apCheckStatusService\_run**

Set to `true`.

## **apPaymentType**

Set to `KLI`.

## **merchantID**

## **merchantReferenceCode**

### **Related information**

[API Field Reference for the Simple Order API](#)

# XML Example: Check Status

This example shows a successful check status request.

## Request

```
<requestMessage
  xmlns="urn:schemas-cybersource-com:transaction-data-1.213">
  <merchantID>test_merch</merchantID>
  <merchantReferenceCode>ref_123</merchantReferenceCode>
  <apPaymentType>KLI</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>6957582387066840103011</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

## Response

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.213">
  <merchantReferenceCode>Postman-1695758700</merchantReferenceCode>
  <requestID>6957587006586637103010</requestID>
  <decision>ACCEPT</decision>
  <reasonCode>100</reasonCode>
  <requestToken>AxjnrwSTeSvqh51d5a+i/6IZYjWmjCbZrR6FeUojmjMjRvSB3i0ZiwyASzejFlof+k3kr2h4vmV
04YwAnSiT</requestToken>
  <apCheckStatusReply>
    <reasonCode>100</reasonCode>
    <reconciliationID>XFZ40MYVGPWJ</reconciliationID>
    <paymentStatus>refunded</paymentStatus>
  </apCheckStatusReply>
</replyMessage>
```

# Name-Value Pair Example: Check Status

This example shows a successful check status request.

## Request

```
apCheckStatusService_run=true
apCheckStatusService_checkStatusRequestID=4848446851386814504011
apPaymentType=KLI
merchantID=mid123
merchantReferenceCode=ref1234
```

## Response

```
merchantReferenceCode=ref1234
requestID=4848446567036715804007
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSTBtW2cINmCVnohjaZCoMrLNy1k07KXFKEb9JpA6cQ1QY9
apCheckStatusReply_reasonCode=100
apCheckStatusReply_paymentStatus=authorized
apCheckStatusReply_reconciliationID=6LBP2Y395ISZ
ap_check_status_payment_status=SETTLED
ap_check_status_rcode=1
ap_check_status_response_code=00004
ap_check_status_rflag=SOK
ap_check_status_rmsg=Request was processed successfully.
ap_check_status_trans_ref_no=TC20000C
ics_rcode=1
ics_rflag=SOK
ics_rmsg=Request was processed successfully.
merchant_ref_number=ref1234
request_id=4865699224970179113680
request_token=AhjnrwSR8oc9Xks3f77JHGENP93fNRxNIJYafra58Mm3AAA8hZW
```

# Authorization Reversals

The authorization reversal service enables you to reverse the amount that was authorized.

## Response Status

The authorization reversal service responds with one of these statuses as the **apAuthReversalReply\_status** field value:

- **AUTH\_REVERSED**: The authorization is successfully reversed.
- **FAILED**: The authorization reversal failed.

The authorization reversal service also responds with a reason code as the **apAuthReversalReply\_reasonCode** field value. For more information on reason codes, see the [Reason Codes for the Simple Order API](#).

## Endpoint

Set the **apAuthReversalService\_run** field to `true` and send the request to one of these endpoints.

- **Production:** <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>
- **Test:** <https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

# Required Fields for Authorization Reversal

Include these required fields to reverse an authorization.

## **apAuthReversalService\_authRequestID**

Set to the request ID included in the authorization response. If you sent a follow-on authorization, use the request ID from the follow-on authorization response.

## **apAuthReversalService\_run**

Set to `true`.

## **apPaymentType**

Set to `KLI`.

## **merchantID**

## **merchantReferenceCode**

### **Related information**

[API Field Reference for the Simple Order API](#)

# XML Example: Authorization Reversal

This example shows a successful authorization reversal.

## Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantID>merch_test</merchantID>
<merchantReferenceCode>ref-123456</merchantReferenceCode>
<apPaymentType>KLI</apPaymentType>
<apAuthReversalService run="true">
  <authRequestID>6958688084846249104011</authRequestID>
</apAuthReversalService>
</requestMessage>
```

## Response

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantReferenceCode>Postman-1695869343</merchantReferenceCode>
<requestID>6958693435966100704008</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>AxjnrwSTeTtFW0Xx4YsI/6IZYjWmjCbZrSIzaQojmjjs01PSBX0GY0MmkmXoxZaH/pN50zJYdUP5
0osAAFAIu</requestToken>
<purchaseTotals>
  <currency>USD</currency>
</purchaseTotals>
<apAuthReversalReply>
  <reasonCode>100</reasonCode>
  <status>AUTH_REVERSED</status>
  <processorResponse>00007</processorResponse>
  <amount>10.00</amount>
  <dateTime>2023-09-28T02:49:04Z</dateTime>
  <paymentStatus>auth_reversed</paymentStatus>
  <responseCode>00007</responseCode>
  <reconciliationID>XFZ40MYVHF6H</reconciliationID>
</apAuthReversalReply>
</replyMessage>
```

# Name-Value Pair Example: Authorization Reversal

This example shows a successful authorization reversal.

## Request

```
apAuthReversalService_run=true
apAuthReversalService_authRequestID=4848446851386814504011
apPaymentType=KLI
merchantID=mid123
merchantReferenceCode=ref1234
```

## Response

```
merchantReferenceCode=ref1234
requestID=4848446567036715804007
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSTBtW2cINmCVnohjaZCoMrLNy1k07KXFKEb9JpA6cQ1QY9
apAuthReversalReply_reasonCode=100
apAuthReversalReply_status=AUTH_REVERSED
apAuthReversalReply_processorResponse=00007
apAuthReversalReply_amount=19.99
apAuthReversalReply_paymentStatus=auth_reversed
apAuthReversalReply_responseCode=00007
apAuthReversalReply_reconciliationID=6LBP2Y395IT0
ap_auth_request_id=4865699224970179113680
ap_payment_type=KLI
ics_applications=ics_ap_auth_reversal
merchant_id=mid12345
merchant_ref_number=ref1234
ap_auth_reversal_amount=19.99
ap_auth_reversal_payment_status=AUTH_REVERSED
ap_auth_reversal_processor_response=00007
ap_auth_reversal_rflag=SOK
ap_auth_reversal_rcode=1
ap_auth_reversal_rmsg=Request was processed successfully.
ap_auth_reversal_response_code=00007
ap_auth_reversal_trans_ref_no=6LBP2Y395ISZ
ics_rcode=1
ics_rmsg=Request was processed successfully.
ics_rflag=SOK
```

merchant\_ref\_number=ref1234  
request\_token=AhjnrwSTB8UkMYfw//LQohjhgxN7L0DLcVqaswikhjD+H/5aAAAygTN  
request\_id=4865699224970179113680

# Captures

The capture service enables you to capture the entire authorized amount or part of the authorized amount. Klarna supports multiple capture requests when the total amount of all captures is less than the authorized amount.

You can capture an authorization for up to 28 days after a payment is authorized.

## Status Responses

The capture service responds with one of these statuses as the **apCaptureReply\_status** response field value:

- **FAILED**: The capture request failed.
- **PENDING**: The capture request is accepted but is not captured. Request the check status service to retrieve status updates. For more information, see Check Status.
- **SETTLED**: The capture request is settled for the requested amount.

The capture service also responds with a reason code as the **apCaptureReply\_reasonCode** field value. For more information on reason codes, see the [Reason Codes for the Simple Order API](#).

## Endpoint

Set the **apCaptureService\_run** field to `true` and send the request to one of these endpoints.

- **Production**: <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>
- **Test**: <https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

# Required Fields for a Capture

Include these required fields to capture an authorized payment.

## **apCaptureService\_authRequestID**

Set to the request ID included in the authorization response.

## **apCaptureService\_run**

Set to `true`.

## **apPaymentType**

Set to `KLI`.

## **merchantID**

## **merchantReferenceCode**

## **purchaseTotals\_currency**

## **purchaseTotals\_grandTotalAmount**

### **Related information**

[API Field Reference for the Simple Order API](#)

# Optional Fields for Captures

Choose from these optional fields to include additional information when capturing a payment.

## **item\_#\_productName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_quantity**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_totalAmount**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_unitPrice**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **Related information**

[API Field Reference for the Simple Order API](#)

# XML Example: Capture

This example shows a successful capture request.

## Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantID>test_merch</merchantID>
<merchantReferenceCode>ref_123456</merchantReferenceCode>
<purchaseTotals>
<currency>USD</currency>
<grandTotalAmount>154.50</grandTotalAmount>
</purchaseTotals>
<apPaymentType>KLI</apPaymentType>
<apCaptureService run="true">
<authRequestID>6957570615206883603012</authRequestID>
</apCaptureService>
</requestMessage>
```

## Response

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantReferenceCode>Postman-1695757617</merchantReferenceCode>
<requestID>6957576170986768803011</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>AxjnrwSTeSvECLGs0aTD/6IZYjWmrWbZoV7NKCojmjMT0vSB3i0ZiwyASZejFlof+k3krsEu
93k+yRAAAy04</requestToken>
<purchaseTotals>
<currency>USD</currency>
</purchaseTotals>
<apCaptureReply>
<reasonCode>100</reasonCode>
<status>SETTLED</status>
<processorResponse>00004</processorResponse>
<amount>154.50</amount>
<dateTime>2023-09-26T19:46:58Z</dateTime>
<reconciliationID>XFZ55MYPWYRA</reconciliationID>
<paymentStatus>settled</paymentStatus>
<responseCode>00004</responseCode>
</apCaptureReply>
</replyMessage>
```

# Name-Value Pair Example: Capture

This example shows a successful capture request.

## Request

```
apCaptureService_run=true
apCaptureService_authRequestID=4848446851386814504011<
apPaymentType=KLI
item_0_unitPrice=19.99
item_0_quantity=1
item_0_productName=Green Widget
item_0_totalAmount=19.99
merchantID=mid123
merchantReferenceCode=ref1234
purchaseTotals_currency=EUR
purchaseTotals_grandTotalAmount=19.99
```

## Response

```
merchantReferenceCode=ref1234
requestID=4848446567036715804007
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSTBtW2cINmCVnohjaZCoMrLNy1k07KXFKEb9JpA6cQ1QY9
purchaseTotals_currency=EUR
apCaptureReply_reasonCode=100
apCaptureReply_processorTransactionID=1217697c-fc8a-4fcc-a13d-20fa83d
    bce98
apCaptureReply_status=SETTLED
apCaptureReply_processorResponse=00004
apCaptureReply_amount=19.99
apCaptureReply_paymentStatus=SETTLED
apCaptureReply_reconciliationID=80G6DY3AKNB7
apCaptureReply_responseCode=00004
ap_payment_type=KLI
ap_auth_request_id=4865699224970179113680
currency=EUR
grand_total_amount=19.99
ics_applications=ics_ap_capture
merchant_id=mid123
merchant_ref_number=ref1234
ap_capture_amount=19.99
ap_capture_status=SETTLED
ap_capture_payment_status=SETTLED
```

```
ap_capture_rflag=SOK
ap_capture_rcode=1
ap_capture_rmsg=Request was processed successfully.
ap_capture_response_code=00007
ap_capture_trans_ref_no=6LBP2Y395ISZ
ap_capture_processor_response=00004
ap_capture_processor_transaction_id=1211697c-fc8a-4fcc-a13d-20fa83
    dbce44
currency=EUR
ics_rcode=1
ics_rmsg=Request was processed successfully.
ics_rflag=SOK
merchant_ref_number=ref1234
request_token=AhjnrwSTB8UkMYfw//LQohjhg
xnN7LODLcVqaswikhjD+H/5aAAAAygTN
request_id=4865699224970179113680
```

# Refunds

The refund service enables you to refund the entire captured amount or part of the captured amount. Klarna supports multiple refund requests when the total amount of all refunds is less than the captured amount.<sup>1</sup>

Cybersource recommends using the optional fields for a refund to provide the best service for the customer. For more information, see [Optional Fields for Refunds \(on page 73\)](#).

## Status Responses

The refund service responds with one of these statuses as the **apRefundReply\_status** field value:

- **FAILED**: The refund request failed.
- **PENDING**: The refund request is accepted but is not refunded. Request the check status service to retrieve status updates. For more information, see [Check Status](#).
- **REFUNDED**: The captured payment is successfully refunded.

The refund service also responds with a reason code as the **apRefundReply\_reasonCode** field value. For more information on reason codes, see the [Reason Codes for the Simple Order API](#).

## Endpoint

Set the **apRefundService\_run** field to `true` and send the request to one of these endpoints.

- **Production**: <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>
- **Test**: <https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

# Required Fields for a Refund

Include these required fields to

## **apPaymentType**

Set to [KLI](#).

## **apRefundService\_refundRequestID**

Set to the request ID included in the capture response.

## **apRefundService\_run**

Set to [true](#).

## **merchantID**

## **merchantReferenceCode**

## **purchaseTotals\_currency**

## **purchaseTotals\_grandTotalAmount**

### **Related information**

[API Field Reference for the Simple Order API](#)

# Optional Fields for Refunds

Choose from these optional fields to include additional information when refunding a payment.

## **billTo\_email**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **billTo(firstName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **billTo(lastName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_productName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_quantity**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_totalAmount**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **item\_#\_unitPrice**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

## **Related information**

[API Field Reference for the Simple Order API](#)

# XML Example: Refund

This example shows a successful refund.

## Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantID>test_merch</merchantID>
<merchantReferenceCode>ref_123456</merchantReferenceCode>
<purchaseTotals>
<currency>USD</currency>
<grandTotalAmount>154.50</grandTotalAmount>
</purchaseTotals>
<apPaymentType>KLI</apPaymentType>
<apRefundService run="true">
<refundRequestID>6957576170986768803011</refundRequestID>
</apRefundService>
</requestMessage>
```

## Response

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.213">
<merchantReferenceCode>Postman-1695758239</merchantReferenceCode>
<requestID>6957582387066840103011</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>AxjnrwSTeSvaHi+ZXThj/6IZYjWmrWbZoV7NKCojmjMcX7SB3i0ZiwyASzejFlof+k3krsEu
93k+yRAABS1G</requestToken>
<purchaseTotals>
<currency>USD</currency>
</purchaseTotals>
<apRefundReply>
<reasonCode>100</reasonCode>
<transactionID>b30d7de1-215d-4989-8b82-f190d39457a4</transactionID>
<status>REFUNDED</status>
<processorResponse>00006</processorResponse>
<amount>154.50</amount>
<dateTime>2023-09-26T19:57:19Z</dateTime>
<reconciliationID>XFZ55MYPWYRA</reconciliationID>
<returnRef>XFZ40MYVGPWJ</returnRef>
<paymentStatus>refunded</paymentStatus>
<responseCode>00006</responseCode>
</apRefundReply>
</replyMessage>
```

# Name-Value Pair Example: Refund

This example shows a successful refund.

## Request

```
apRefundService_run=true
apRefundService_refundRequestID=4848447150406200004010
apPaymentType=KLI
billTo(firstName=Anna
billTo(lastName=Schmidt
billTo_email=schmidt@example.com
item_0_unitPrice=19.99
item_0_quantity=1
item_0_productName=Green Widget
item_0_totalAmount=19.99
merchantID=mid123
merchantReferenceCode=ref1234
purchaseTotals_currency=EUR
purchaseTotals_grandTotalAmount=19.99
```

## Response

```
merchantReferenceCode=ref1234
requestID=4848446567036715804007
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSTBtW2cINmCVnohjaZCoMrLNy1k07KXFKEb9JpA6cQ1QY9
purchaseTotals_currency=EUR
apRefundReply_reasonCode=100
apRefundReply_processorTransactionID=4cbe83f2-5deb-4535-b534-de9b7099
    dd50
apRefundReply_status=REFUNDED
apRefundReply_processorResponse=00006
apRefundReply_amount=19.99
apRefundReply_reconciliationID=72WTZY39N3YT
apRefundReply_paymentStatus=REFUNDED
apRefundReply_responseCode=00006
```

# Reason Codes and Klarna Response Codes

Cybersource recommends using the reason code value and the Klarna response code to determine the transaction result.

## Reason Codes and Klarna Response Codes

Reason Code	Klarna Response Codes	Description
100	<ul style="list-style-type: none"><li>• <a href="#">00000</a>: Completed</li><li>• <a href="#">00001</a>: Pending</li><li>• <a href="#">00002</a>: Abandoned</li><li>• <a href="#">00003</a>: Authorized</li><li>• <a href="#">00004</a>: Settled</li><li>• <a href="#">00006</a>: Refunded</li></ul>	Successful transaction.
102	<a href="#">10000</a> : Failed	One or more fields in the request contain invalid data.
150	<ul style="list-style-type: none"><li>• <a href="#">20000</a>: Failed</li><li>• <a href="#">20001</a>: Failed</li><li>• <a href="#">20002</a>: Failed</li><li>• <a href="#">30000</a>: Failed</li><li>• <a href="#">30100</a>: Failed</li></ul>	Possible reasons: <ul style="list-style-type: none"><li>• The signature was not included in the HTTP header.</li><li>• The signature in the HTTP header has expired, or it is not a valid signature.</li><li>• The API version in the HTTP header was missing or is not supported.</li></ul>
203	<ul style="list-style-type: none"><li>• <a href="#">30200</a>: Failed</li><li>• <a href="#">30400</a>: Failed</li><li>• <a href="#">30500</a>: Failed</li></ul>	Klarna declined the transaction because of funding source problems, or the transaction was flagged as high risk.
204	<a href="#">30350</a> : Failed	Payment declined because of insufficient funds in the account.
233	<ul style="list-style-type: none"><li>• <a href="#">30600</a>: Failed</li><li>• <a href="#">30700</a>: Failed</li></ul>	Klarna declined the transaction because of tax errors or government compliance errors.

# Reporting

You can generate different types of reports for your financial and reconciliation data. For more information on how to generate these reports, see the *Reporting User Guide*.

The *Reporting User Guide* contains these relevant topics:

- How and When Reports Are Generated
- Downloading Available Reports
- Subscribing to Standard Reports

## Additional Resources

### Business Center Overview

For an overview of the various resources available in the Business Center, see this YouTube video:

<https://www.youtube.com/watch?v=UDmAWGHPbWs>

### Business Center Navigation

For a step-by-step demonstration of how to navigate in the Business Center, see this YouTube video:

[https://www.youtube.com/watch?v=2qi\\_g2DParI](https://www.youtube.com/watch?v=2qi_g2DParI)

### Managing Report Subscriptions

For an overview of how to manage report subscriptions in the Downloadable Reports section in the Business Center, see this YouTube video:

<https://www.youtube.com/watch?v=tFlmkXtvxWE>

### Downloading Reports

For an overview of how to download available reports in the Reports section in the Business Center, see this YouTube video:

<https://www.youtube.com/watch?v=E0slUYjJvmw>

# Simple Order API Field Map

## Field Map

Klarna API Field Name	Simple Order API Field Name
authorization_token	apAuthService_preapprovalToken
billing_address.city	billTo_city
billing_address.country	billTo_country
billing_address.email	billTo_email
billing_address.family_name	billTo_lastName
billing_address.given_name	billTo(firstName)
billing_address.phone	billTo_phoneNumber
billing_address.postal_code	billTo_postalCode
billing_address.region	billTo_district
billing_address.street_address	billTo_street1
billing_address.street_address2	billTo_street2
billing_address.title	billTo_title
customer.date_of_birth	billTo_dateOfBirth
customer.gender	billTo_gender
locale	billTo_language
merchant_urls.confirmation	apAuthService_successURL
options.color_border	apUI_colorBorder
options.color_border_selected	apUI_colorBorderSelected
options.color_button	apUI_colorButton
options.color_button_text	apUI_colorButtonText
options.color_checkbox	apUI_colorCheckbox
options.color_checkbox_checkmark	apUI_colorCheckboxCheckMark
options.color_header	apUI_colorHeader
options.color_link	apUI_colorLink
options.color_text	apUI_colorText
options.payment_method_category	paymentMethod_name

**Field Map (continued)**

Klarna API Field Name	Simple Order API Field Name
options.radius_border	apUI_borderRadius
order_amount	purchaseTotals_grandTotalAmount
order_lines.name	item__productName
order_lines.reference	item__productSKU
order_lines.tax_rate	item__taxRate
order_lines.total_amount	item__totalAmount
order_lines.unit_price	item__unitPrice
order_lines.quantity	item__quantity
order_lines.tax_amount	item__taxAmount
purchase_country	billTo_country
purchase_currency	purchaseTotals_currency
shipping_address.city	shipTo_city
shipping_address.country	shipTo_country
shipping_address.email	shipTo_email
shipping_address.family_name	shipTo_lastName
shipping_address.given_name	shipTo(firstName)
shipping_address.phone	shipTo_phoneNumber
shipping_address.postal_code	shipTo_postalCode
shipping_address.region	shipTo_district
shipping_address.street_address	shipTo_street1
shipping_address.street_address2	shipTo_street2
shipping_address.title	shipTo_title