Payments

REST API
Visa Platform Connect

cybersource
A Visa Solution

Developer Guide

Cybersource Contact Information

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

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

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

Copyright

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 suparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in Cybersource Corporation's standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

Trademarks

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

Confidentiality Notice

This document is furnished to you solely in your capacity as a client of Cybersource and as a participant in the Visa payments system.

By accepting this document, you acknowledge that the information contained herein (the "Information") is confidential and subject to the confidentiality restrictions contained in Visa's operating regulations and/or other confidentiality agreements, which limity our use of the Information. You agree to keep the Information confidential and not to use the Information for any purpose other than its intended purpose and in your capacity as a customer of Cybersource or as a participant in the Visa payments system. The Information may only be disseminated within your organization on a need-to-know basis to enable your participation in the Visa payments system. Please be advised that the Information may constitute material non-public information under U.S. federal securities laws and that purchasing or selling securities of Visa Inc. while being aware of material non-public information would constitute a violation of applicable U.S. federal securities laws.

Revision

25.07.01

Contents

Payments Developer Guide

This section describes 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 REST API to integrate payment card processing into an order management system.

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

Conventions

These statements appear 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

Visit the *Cybersource documentation hub* to find additional processor-specific

	versions of this guide and additional technical documentation.
Customer Support	**For support information about any service, visit the Support Center:** *http://support.visaacceptance.com*

Recent Revisions to This Document

25.07.01

PayPak Cards	**Added PayPak cards. See** *Card Types* **on page 27.**

25.05.01

International Transaction Compliance	**Added a section about international transaction compliance. See** *Compliance* **on page 30.**

25.04.01

This revision contains only editorial changes and no technical updates.

25.03

Mass Transit Processing	**Added new EMV and card data information for payment services. See** *Mass Transit Payment Services Using EMV and Card Data* **on page 152.**

25.02

Incremental Authorizations	**Added an optional field. See** *Optional Field for Processing an Incremental Authorization* **on page 95.**
Retail Processing	**Added new EMV and card data information for payment services. See** *Retail EMV and Card Data* **on page 128.** **Updated fields and examples. See** *Authorization for Cash Advance with Credit Card* **on page 145.**
Level II and Level III Processing	**Updated the examples for Level II and Level III captures. See** *Captures with Level II Data*

	on page 251 and *Captures with Level III Data* **on page 261.**

25.01

Added a testing section. See *Testing the Payment Services* on page 55.

Marketplace Transactions with Foreign Retailers	**Added authorizations and captures. See these sections:** • *Marketplace Authorizations with Foreign Retailers* on page 76 • *Marketplace Captures with Foreign Retailers* on page 113
Mass Transit Processing	**Added support for American Express cards. See these sections:** • *American Express Account Status Check Authorization with EMV Data* on page 154 • *American Express Delayed Online Authorization with EMV Data* on page 157 **Added support for Discover cards in the U.S. See these sections:** • *Discover Authorization with EMV Data* on page 161 • *Discover Sale with EMV Data* on page 171
Credentialed Transactions	**Removed Mastercard required field for retrieving customer credentials during a CIT request. See** *Card-Specific Required Field for Retrieving Customer Credentials During a CIT* **on page 319.** **Added new country-specific fields for requesting a subsequent installment payment. See** *Country-Specific Required Fields for Installment Payments with Mastercard or Visa Card* **on page 401.**

24.14

This revision contains only editorial changes and no technical updates.

24.13

This revision contains only editorial changes and no technical updates.

24.12

This revision contains only editorial changes and no technical updates.

24.11

This revision contains only editorial changes and no technical updates.

24.10

This revision contains only editorial changes and no technical updates.

24.09

This revision contains only editorial changes and no technical updates.

24.08

This revision contains only editorial changes and no technical updates.

VISA Platform Connect: Specifications and Conditions for Resellers/Partners

The following are specifications and conditions that apply to a Reseller/Partner enabling its merchants through Cybersource for Visa Platform Connect ("VPC") processing. Failure to meet any of the specifications and conditions below is subject to the liability provisions and indemnification obligations under Reseller/Partner's contract with Visa/Cybersource.

1. Before boarding merchants for payment processing on a VPC acquirer's connection, Reseller/Partner and the VPC acquirer must have a contract or other legal agreement that permits Reseller/Partner to enable its merchants to process payments with the acquirer through the dedicated VPC connection and/or traditional connection with such VPC acquirer.
2. Reseller/Partner is responsible for boarding and enabling its merchants in accordance with the terms of the contract or other legal agreement with the relevant VPC acquirer.
3. Reseller/Partner acknowledges and agrees that all considerations and fees associated with chargebacks, interchange downgrades, settlement issues, funding delays, and other processing related activities are strictly between Reseller and the relevant VPC acquirer.
4. Reseller/Partner acknowledges and agrees that the relevant VPC acquirer is responsible for payment processing issues, including but not limited to, transaction declines by network/issuer, decline rates, and interchange qualification, as may be agreed to or outlined in the contract or other legal agreement between Reseller/Partner and such VPC acquirer.

DISCLAIMER: NEITHER VISA NOR CYBERSOURCE WILL BE RESPONSIBLE OR LIABLE FOR ANY ERRORS OR OMISSIONS BY THE Visa Platform Connect ACQUIRER IN PROCESSING TRANSACTIONS. NEITHER VISA NOR CYBERSOURCE WILL BE RESPONSIBLE OR LIABLE

FOR RESELLER/PARTNER BOARDING MERCHANTS OR ENABLING MERCHANT PROCESSING IN VIOLATION OF THE TERMS AND CONDITIONS IMPOSED BY THE RELEVANT Visa Platform Connect ACQUIRER.

Introduction to Payments

This introduction provides the basic information that you need to successfully process payment transactions. It also provides an overview of the payments industry and provides workflows for each process.

With Cybersource payment services, you can process payment cards (tokenized or non-tokenized), digital payments such as Apple Pay and Google Pay, and customer ID transactions. You can process payments across the globe and across multiple channels with scalability and security. Cybersource supports a large number of payment cards and offers a wide choice of gateways and financial institutions, all through one connection.

Visit the *Cybersource documentation hub* to find additional processor-specific versions of this guide and additional technical documentation.

Financial Institutions and Payment Networks

Financial institutions and payment networks enable payment services to function. These entities work together to complete the full payment cycle.

Merchant Financial Institutions (Acquirers)

A merchant financial institution, also known as an acquirer, offers accounts to businesses that accept payment cards. Before you can accept payments, you must have a merchant account from an acquirer. Your merchant account must be configured to process card-not-present, card-present, or mail-order/telephone-order (MOTO) transactions.

Each acquirer has connections to a limited number of payment processors. You must choose a payment processor that your acquirer supports.

You can expect your acquirer to charge these fees:

- Discount rates: your acquirer charges a fee and collects a percentage of every transaction. The combination of the fee and the percentage is called the discount rate. These charges can be bundled (combined into a single charge) or unbundled (charged separately).

- Interchange fees: payment networks, such as Visa or Mastercard, each have a base fee, called the interchange fee, for each type of transaction. Your acquirer and processor can show you ways to reduce this fee.
- Chargebacks: when cardholders dispute charges, you can incur chargebacks. A chargeback occurs when a charge on a customer's account is reversed. Your acquirer removes the money from your account and could charge you a fee for processing the chargeback.

Take these precautions to prevent chargebacks:

- Use accurate merchant descriptors so that customers can recognize the transactions on their statements.
- Provide good customer support.
- Ensure rapid problem resolution.
- Maintain a high level of customer satisfaction.
- Minimize fraudulent transactions.

If excessive chargebacks or fraudulant changes occur, these actions might be taken:

- You might be required to change your business processes to reduce the number chargebacks, fraud, or both.
- Your acquiring institution might increase your discount rate.
- Your acquiring institution might revoke your merchant account.

Contact your sales representative for information about products that can help prevent fraud.

Customer Financial Institutions (Issuers)

A customer financial institution, also known as an issuer, provides payment cards to and underwrites lines of credit for their customers. The issuer provides monthly statements and collects payments. The issuer must follow the rules of the payment card companies to which they belong.

Payment Networks

Payment networks manage communications between acquiring financial institutions and issuing financial institutions. They also develop industry standards, support their brands, and establish fees for acquiring institutions.

Some payment networks, such as Visa, Mastercard, and UnionPay International, are trade associations that do not issue cards. Issuers are members of these associations, and they issue cards under license from the association.

Other networks, such as Discover and American Express, issue their own cards. Before you process cards from these companies, you must sign agreements with them.

Payment Processors

Payment processors connect with acquirers. Before you can accept payments, you must register with a payment processor. An acquirer might require you to use a payment processor with an existing relationship with the acquirer.

Your payment processor assigns one or more merchant IDs (MIDs) to your business. These unique codes identify your business during payment transactions.
This table lists the processors and corresponding card types that are supported for payment services.

> **Important**
>
> Only the card types explicitly listed here are supported.

Payment Processors and Supported Card Types

Payment Processor	Supported Card Types	Notes
Visa Platform Connect	Different card types are supported for each Visa Platform Connectacquirer. See Visa Platform Connect Acquirers.	The Visa Electron card type is processed the same way that the Visa debit card is processed. Use card type value 001 (Visa) for Visa Electron.

Visa Platform Connect Acquirers

The following acquirers and card types are supported for Visa Platform Connect:

Supported Visa Platform Connect Acquirers and Card Types

Raw Processor Name	Processor Name	Supported Card Types
vdcabsa	Absa Bank on Visa Platform Connect	Visa, Mastercard, JCB, Diners Club
vdcagbkchina	Agricultural Bank of China (ABC) on Visa Platform Connect	Visa, Mastercard, American Express, JCB, Diners Club Visa Platform Connect cannot process domestic transactions in China. Visa Platform Connect can process only cross-border transactions. A crossborder transaction is a transaction for which the payment card is issued in another country and accepted by a merchant in China.
networkintluae	Ahli United Bank in Bahrain, BLOM Bank, Network International	Visa, Mastercard, JCB, Diners Club

Raw Processor Name	Processor Name	Supported Card Types
vdcaaib	Arab African International Bank (AAIB) on Visa Platform Connect	Visa, Mastercard, JCB
vdcacbvietnam	Asia Commercial Bank (ACB) on Visa Platform Connect	Visa, Mastercard, JCB
vdcasb	Auckland Savings Bank (ASB) on Visa Platform Connect	Visa, Mastercard
vdcanzbank	Australia and New Zealand Banking Group Ltd. (ANZ) on Visa Platform Connect	Visa, Mastercard
vdcaxis	Axis Bank Ltd. of India on Visa Platform Connect	Visa, Mastercard, Diners Club
vdcbanamex	Banco Nacional de México (Banamex) on Visa Platform Connect	Visa, Mastercard, American Express, Discover, JCB, Diners Club
vdcbcosafrabr	Banco Safra on Visa Platform Connect	Visa, Mastercard, American Express
vdcbbl	Bangkok Bank Ltd. on Visa Platform Connect	Visa, Mastercard, JCB
vdcbankmuscat	Bank Muscat of Oman on Visa Platform Connect	Visa, Mastercard, American Express, Diners Club
vdcbay	Bank of Ayudhya (BAY) on Visa Platform Connect	Visa, Mastercard, JCB
vdcbocmacau	Bank of China in Macau on Visa Platform Connect	Visa, Mastercard
vdcbocom	Bank of Communications on Visa Platform Connect	Visa, Mastercard
vdcbksinarmasid	Bank Sinarmas (Omise Ltd.) on Visa Platform Connect	Visa, Mastercard

Raw Processor Name	Processor Name	Supported Card Types
vdcbcellao	Banque Pour Le Commerce Exterieur Lao (BCEL) on Visa Platform Connect	Visa, Mastercard, American Express, JCB
vdcbarclaysbw	Barclays Bank Botswana on Visa Platform Connect	Visa, Mastercard, American Express
vdcbarclaysmu	Barclays Bank Mauritius Ltd. on Visa Platform Connect	Visa, Mastercard, American Express
vdcbarclaysghtzug	Barclays Bank of Ghana Ltd., Barclays Bank of Tanzania Ltd., and Barclays Bank of Uganda Ltd. on Visa Platform Connect	Visa, Mastercard, American Express
vdcbarclayske	Barclays Bank of Kenya on Visa Platform Connect	Visa, Mastercard, American Express
vdcbarclayszm	Barclays Bank of Zambia on Visa Platform Connect	Visa, Mastercard, American Express
vdcbarclayssc	Barclays Bank Seychelles on Visa Platform Connect	Visa, Mastercard, American Express
vdcbccardkr	BC Card Co., Ltd. on Visa Platform Connect	Visa, Mastercard, American Express, JCB
vdccubtw	Cathay United Bank (CUB) on Visa Platform Connect	Visa, Mastercard, JCB
vdccitihkmo	Citibank Hongkong and Macau on Visa Platform Connect	Visa, Mastercard, Diners Club, JCB
vdccitimy	Citibank Malaysia on Visa Platform Connect	Visa, Mastercard
vdccitisg	Citibank Singapore Ltd. on Visa Platform Connect	Visa, Mastercard, JCB
vdccbq	Commercial Bank of Qatar on Visa Platform Connect	Visa, Mastercard, American Express, JCB, Diners Club
vdccredimax	CrediMax (Bahrain) on Visa Platform Connect	Visa, Mastercard, American Express, JCB, Diners Club

Raw Processor Name	Processor Name	Supported Card Types
vdcctbc	CTBC Bank Ltd. on Visa Platform Connect	Visa, Mastercard, JCB
vdcfdmsbn	First Data Merchant Solutions in Brunei on Visa Platform Connect	Visa, Mastercard, JCB
vdcfdmshk	First Data Merchant Solutions in Hong Kong on Visa Platform Connect	Visa, Mastercard, JCB
vdcfdmsmy	First Data Merchant Solutions in Malaysia on Visa Platform Connect	Visa, Mastercard, JCB
vdcfdmssg	First Data Merchant Solutions in Singapore on Visa Platform Connect	Visa, Mastercard, JCB
vdcfnb	FirstRand Bank on Visa Platform Connect	Visa, Mastercard, American Express, Diners Club
vdchsbcbank	Global Payments Asia Pacific on Visa Platform Connect	Visa, Mastercard, JCB In India, the only supported card types are Visa and Mastercard. All three card types (Visa, Mastercard, JCB) are supported in all other countries that Global Payments Asia Pacific covers.
vdchabibltd	Habib Bank Ltd. (HBL) on Visa Platform Connect	Visa, Mastercard, American Express, JCB, Diners Club
vdchdfc	HDFC Bank Ltd. of India on Visa Platform Connect	Visa, Mastercard, Diners Club
vdcimbank	I&M Bank on Visa Platform Connect	Visa, Mastercard
vdcicici	ICICI of India on Visa Platform Connect	Visa, Mastercard

Raw Processor Name	Processor Name	Supported Card Types
vdckeb	Korea Exchange Bank (KEB) on Visa Platform Connect	Visa, Mastercard, JCB Visa Platform Connect cannot process domestic transactions in Korea. Visa Platform Connect can process only cross-border transactions. A crossborder transaction is a transaction for which the payment card is issued in another country and accepted by a merchant in Korea.
vdcmashreqbk	Mashreq on Visa Platform Connect	Visa, Mastercard, American Express, JCB, Diners Club
vdcmaybankmy	Maybank on Visa Platform Connect	Visa, Mastercard, American Express, JCB
vdcnbad	National Bank of Abu Dhabi (NBAD) on Visa Platform Connect	Visa, Mastercard, JCB, Diners Club
vdcnbk	National Bank of Kuwait (NBK) on Visa Platform Connect	Visa, Mastercard, Diners Club
vdcnacombk	National Commercial Bank on Visa Platform Connect	Visa, Mastercard, mada
vdcnijo	Network International (NI) Jordan on Visa Platform Connect	Visa, Mastercard, American Express, JCB, Diners Club
vdcocbc	Overseas Chinese Banking Corp (OCBC) on Visa Platform Connect	Visa, Mastercard
vdcpromerica	Promerica in Honduras and Nicaragua on Visa Platform Connect	Visa, Mastercard
vdcbni	PT Bank Negara Indonesia on Visa Platform Connect	Visa, Mastercard
vdcqnbqa	Qatar National Bank (QNB Group) on Visa Platform Connect	Visa, Mastercard, American Express, JCB, Diners Club

Raw Processor Name	Processor Name	Supported Card Types
vdcsacomb	Sacombank on Visa Platform Connect	Visa, Mastercard, JCB
vdcsmcc	Sumitomo Mitsui Card Co. on Visa Platform Connect	Visa
vdctaishintw	Taishin Bank Ltd. on Visa Platform Connect	Visa, Mastercard, American Express, JCB
vdcuob	United Overseas Bank (UOB) in Singapore and Vietnam on Visa Platform Connect	Visa, Mastercard, JCB
vdcuobth	United Overseas Bank (UOB) in Thailand on Visa Platform Connect	Visa, Mastercard
vdcvantiv	Vantiv on Visa Platform Connect	Visa, Mastercard, American Express, Discover, JCB, Diners Club
vdcvietcombk	Vietcombank on Visa Platform Connect	Visa, Mastercard, American Express, JCB, Diners Club
vdcvietin	VietinBank on Visa Platform Connect	Visa, Mastercard, JCB, Diners Club
vdctechcomvn	Vietnam Technological and Commercial Joint Stock Bank (Techcombank) on Visa Platform Connect	Visa, Mastercard, American Express, JCB, Diners Club
vdcguatemala	Visa Guatemala on Visa Platform Connect	Visa
vdcvisanetuy	VisaNet Uruguay on Visa Platform Connect	Visa
vdcwestpac	Westpac on Visa Platform Connect	Visa, Mastercard
vdcwhb	Wing Hang Bank on Visa Platform Connect	Visa, Mastercard
vdcwinglung	Wing Lung Bank on Visa Platform Connect	Visa, Mastercard

Card Types

You can process payments with these kinds of cards:

- Co-badged cards
- Co-branded cards
- Credit cards
- Debit cards
- Prepaid cards
- Private label cards
- Quasi-cash

You can process payments with these card types:

- American Express
- China UnionPay
- Diners Club
- Discover
- JCB
- Mastercard
- Meeza (Pilot in Egypt only)
- PayPak
- Visa

Co-Badged Cards

Co-badged cards are credit and debit cards that integrate two or more payment networks.

mada Co-Badged Cards

mada is Saudi Arabia's domestic payment network.
These mada co-badged debit cards are supported:

- Visa and mada
- Mastercard and mada

mada co-badged debit cards are processed as follows:

- Only domestic processing is supported in Saudi Arabia.
- Transactions are sent directly to the Saudi Arabia Monetary Authority (SAMA) for processing.
- Payer authentication is supported. Visa Secure is supported for co-badged Visa and mada cards. Mastercard Identity Check is supported for co-badged Mastercard and mada cards.
- For acquirers, the card type is identified as MD.
- In reports, the card type is identified as either Visa or Mastercard.

- Dual-message processing is not supported. Only single-message processing is supported.

Co-Branded Cards

Co-branded cards are credit cards that are branded with a merchant's logo, brand, or other identifier as well as the payment network logo. These cards are not limited for use at the branded merchant and can be used at any merchant that accepts credit cards.

Credit Cards

Cardholders use credit cards to borrow money from issuing banks to pay for goods and services offered by merchants that accept credit cards.

Debit Cards

A debit card is linked to a cardholder's checking account. A merchant who accepts the debit card can deduct funds directly from the account.

Prepaid Cards

Prepaid cards enable cardholders to pay for goods and services using money stored directly on the card.

Private Label Cards

Private label cards are issued by private companies. They enable cardholders to borrow money to pay for goods exclusively at the issuing company's stores.

Quasi-Cash

Quasi-cash transactions involve instruments that are directly convertible to cash such as web wallets, travelers checks, cryptocurrency, and lottery tickets.

Transaction Types

This topic provides information about transaction types that are supported by your processor, such as card-present, card-not-present, and international transactions.

Card-Not-Present Transactions

When a customer provides a card number, but the card and the customer are not physically present at the merchant's location, the purchase is known as a card-not-present transaction. Typical card-not-present transactions are internet and phone transactions. Card-not-present transactions pose an additional level of risk to your business because the customer's identification cannot be verified. You can reduce that risk by using features such as the Address Verification System (AVS) and Card Verification

Numbers (CVNs). The AVS and CVNs provide additional protection from fraud by verifying the validity of the customer's information and notifying you when discrepancies occur.

Card-Present Transactions

When a customer uses a card that is physically present in a retail environment, the purchase is known as a card-present transaction.

Authorizations with Card Verification Numbers

Card verification numbers (CVNs) are a required feature for the authorization service. The CVN is printed on a payment card, and only the cardholder can access it. The CVN is used in card-not-present transactions as a verification feature. Using the CVN helps reduce the risk of fraud.

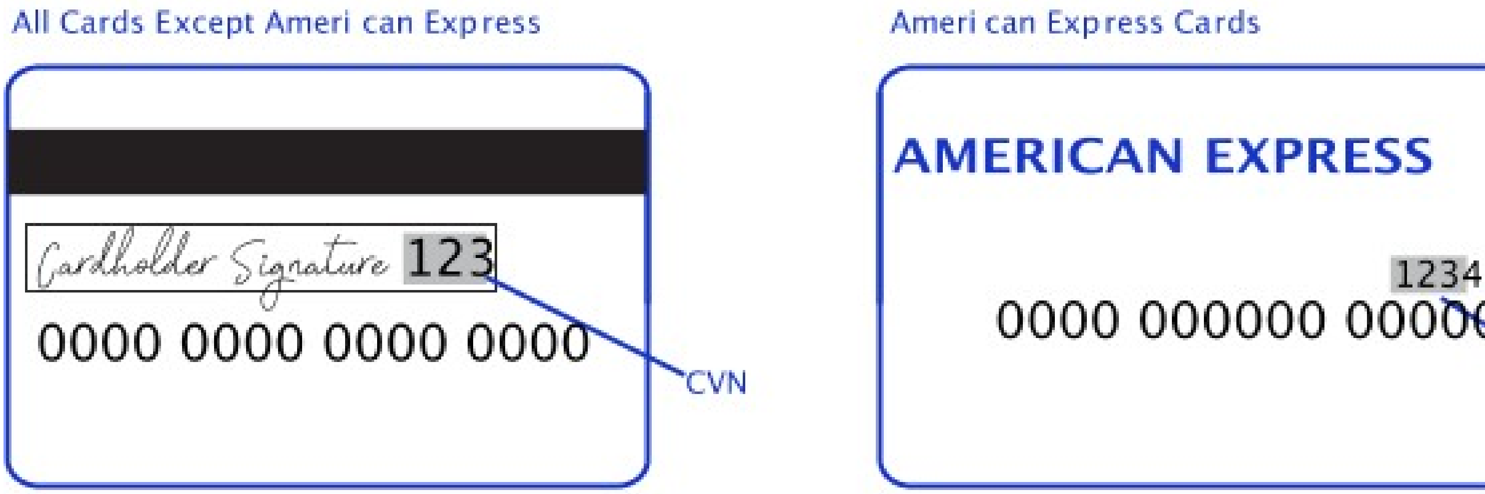
CVNs are not included in payment card track data and cannot be obtained from a card swipe, tap, or dip.

CVNs must not be stored after authorization.

> **Important**
>
> In Europe, Visa mandates that you not include a CVN for mail-order transactions and not record a CVN on any physical format such as a mail-order form.

CVN Locations and Terminology

For most cards, the CVN is a three-digit number printed on the back of the card, to the right of the signature field. For American Express, the CVN is a four-digit number printed on the front of the card above the card number.



CVN Locations

Each payment card company has its own name for the CVN value:

- American Express and Discover call it the Card Identification Number (CID).
- JCB calls it the Card Authentication Value (CAV2).
- Mastercard calls it the Card Validation Code (CVC2).
- Visa calls it the Card Verification Value (CVV2).

International Transactions

Consider compliance and merchant remittance funding when processing international transactions.

Compliance

Accepting payments from a country other than your own requires that you observe the processing rules and practices of the payment systems in that country. This list describes areas of compliance that are especially important:

- Merchant descriptor requirements—A merchant descriptor communicates merchant information to customers to remind them of the circumstances that triggered a payment. Merchant descriptors reduce the possibility of a chargeback. Accordingly, the merchant descriptor displayed on a customer’s statement should be a close match to the name on your website. It is not good practice to consolidate multiple websites into a single merchant account and use a generic descriptor that more-or-less covers all offerings.
- Excessive chargebacks—To prevent an excessive number of chargebacks, you must maintain good customer support, rapid problem resolution, a high level of customer satisfaction, and transaction management processes that minimize fraudulent transactions. When payment card chargebacks become excessive, you must change business processes to reduce chargebacks. If chargebacks are not reduced to a satisfactory level, your account can be terminated.

Merchant Remittance Funding

You can request that the transaction proceeds be converted to another currency. Currency conversion uses a foreign exchange rate to calculate the conversion to the requested currency. The foreign exchange rate might be explicitly stated as a rate or implicitly stated as a transaction amount. The funded amount and can vary from day to day. The foreign exchange rate might also include an increase for the foreign exchange risk, sales commissions, and handling costs.

Token Management Service

The Token Management Service (TMS) tokenizes, securely stores, and manages customer and payment data. TMS enables you to:

- Securely store a customer's payment details and their billing and shipping addresses.
- Create a network token of a customer's payment card.

TMS simplifies your PCI DSS compliance. TMS passes back to you tokens that represent this data. You then store these tokens in your environment and databases instead of customer payment details.

TMS Token Types

- Customer — Stores the buyer’s email address and the merchant's account ID for that buyer plus any other custom fields.
- Shipping Address — Stores a shipping address for a specific customer.
- Instrument Identifier — Stores either a payment card number or a bank account number and routing number
  This resource creates either:
  - An Instrument Identifier token using details of a payment card or an ACH bank account.
  - A payment network token using the details of a payment card; also uses the card expiration date and billing address, which are pass-through only fields.
- Payment Instrument — Stores a Payment Instrument using an Instrument Identifier token. It does not store the card number and cannot exist without an associated Instrument Identifier. It stores:
  - Card expiration date
  - Billing address

  You can also choose to store this information yourself instead and store only the card number or bank account and routing number in an Instrument Identifier object.
- Customer Payment Instrument — Creates and stores a payment instrument for a specific customer ID and an Instrument Identifier token.

TMS Features

- Create, retrieve, update, and delete tokens.
- Set a default payment instrument and shipping address for a customer.
- Process follow-on payment transactions with token IDs.
- Create and update tokens through bundled payment transactions.

Important

Due to mandates from the Reserve Bank of India, Indian merchants cannot store personal account numbers (PAN). Use network tokens instead. For more information on network tokens, see the Network Tokenization section of the *Token Management Service Guide.*

Payment Services

Various services are involved in processing payments.
These services enable customers to purchase goods and services. They also enable merchants to receive payments from customer accounts, to provide refunds, and to void transactions.

Authorizations

An authorization confirms that a payment card account holds enough funds to pay for a purchase. Authorizations can be made online or offline.

Online Authorizations

Online authorizations provide immediate confirmation of funds availability. The customer's financial institution also reduces the amount of credit available in the customer's account, setting aside the authorized funds for the merchant to capture at a later time. Authorizations for most payment cards are processed online. Typically, it is safe to start fulfilling the order when you receive an authorization confirmation.

An online authorization confirmation and the subsequent hold on funds expire after a specific length of time. Therefore it is important to capture funds in a timely manner. The issuing bank sets the expiration time interval, but most authorizations expire within 5 to 7 days.

The issuing bank does not inform Cybersource when an authorization confirmation expires. By default, the authorization information for each transaction remains in the Cybersource database for 180 days after the authorization date. To capture an authorization that expired with the issuing bank, you can resubmit the authorization request.

Offline Authorizations

Online transactions require an internet connection. In situations where the internet is not available, for example, due to an outage, merchants can continue to take credit card payments using offline transactions. An offline authorization is an authorization request for which you do not receive an immediate confirmation about the availability of funds. Offline authorizations have a higher level of risk than online transactions because they do not confirm funds availability or set aside the funds for later capture. Further, it can take up to 5 days to receive payment confirmations for offline transactions. To mitigate this risk, merchants may choose to fulfill orders only after receiving payment confirmation.

Pre-Authorizations

A pre-authorization enables you to authorize a payment when the final amount is unknown. It is typically used for lodging, auto rental, e-commerce, and restaurant transactions.

For a pre-authorization:

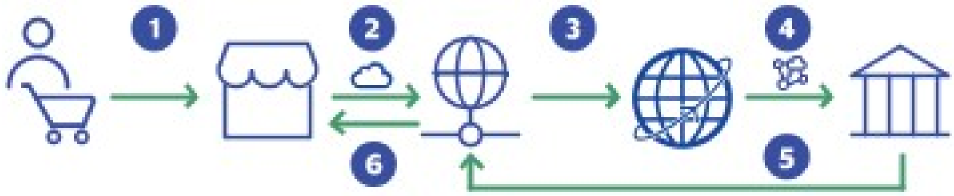
- The authorization amount must be greater than zero.
- The authorization must be submitted for capture within 30 calendar days of its request.
- When you do not capture the authorization, you must reverse it.
  In the U.S., Canada, Latin America, and Asia Pacific, Mastercard charges an additional fee for a pre-authorization that is not captured and not reversed.
  In Europe, Russia, Middle East, and Africa, Mastercard charges fees for all pre-authorizations.
- Chargeback protection is in effect for 30 days after the authorization.

Payment Network Token Authorizations

You can integrate authorizations with payment network tokens into your existing order management system. For an incremental authorization, you do not need to include any payment network tokenization fields in the authorization request because Cybersource obtains the payment network tokenization information from the original authorization request.

Authorization Workflow

This image and description show the authorization workflow:



1. The customer purchases goods or services from the merchant using a payment card.
2. You send an authorization request over secure internet connection to Cybersource. When the customer buys a digitally delivered product or service, you can request both the authorization and the capture at the same time. When the customer buys a physically fulfilled product, do not request the capture until you ship the product.
3. Cybersource validates the order information then contacts your payment processor and requests authorization.
4. The processor sends the transaction to the payment card company, which routes it to the issuing bank for the customer's payment card. Some card companies, including Discover and American Express, act as their own issuing banks.
5. The issuing bank approves or declines the request.
    - If funds are available, the issuing bank reserves the amount of the authorization request and returns an authorization approval to Cybersource.
    - If the issuing bank denies the request, it returns an authorization denial to Cybersource.
6. Cybersource runs its own tests then tells you whether the authorization succeeded.

Sales

A sale is a bundled authorization and capture. Some processors and acquirers require a sale transaction instead of using separate authorization and capture requests. For other processors and acquirers, you can request a sale instead of a separate authorization and capture when you provide the goods or services immediately after taking an order. There are two types of sale processing: dual-message processing and single-message processing.

Dual-Message Processing

Dual-message processing is a two-step process. The authorization is processed first. If the authorization is successful, the capture is processed immediately afterward. The response includes the authorization and the capture information. If the authorization is declined, the capture is not processed, and the response message includes only the authorization information.

Partial Authorizations

All debit and prepaid card processors as well as a limited number of credit card processors support partial authorizations when dual-message processing is in place.

When partial authorization is enabled, the issuing financial institution can approve a partial amount when the balance on the card is less than the requested amount. When a partial amount is authorized, the capture is not processed. The merchant can then use a second card to cover the balance, adjust the total cost, or void the transaction.

Single-Message Processing

Single-message processing treats the authorization and capture as a single transaction. There are important differences between dual-message processing and single-message processing:

- Single-message processing treats the request as a full-financial transaction, and with a successful transaction, funds are immediately transferred from the customer account to the merchant account.
- Authorization and capture amounts must be the same.
- Some features cannot be used with single-message processing.

Authorization Reversals

The authorization reversal service releases the hold that an authorization placed on a customer’s payment card funds.

Each card-issuing financial institution has its own rules for deciding whether an authorization reversal succeeds or fails. When a reversal fails, contact the card-issuing financial institution to learn whether there is a different way to reverse the authorization.

If your processor supports authorization reversal after void (ARAV), you can reverse an authorization after you void the associated capture. If your processor does not support ARAV, you can use the authorization reversal service only for an authorization that has not been captured and settled.

An authorization reversal is a follow-on transaction that uses the request ID returned from an authorization. The main purpose of a follow-on transaction is to link two transactions. The request ID links the follow-on transaction to the original transaction.

The authorization request ID is used to look up the customer’s billing and account information in the Cybersource database. You are not required to include those fields in the full authorization reversal request. The original transaction and follow-on transaction are linked in the database and in the Business Center.

For processors that support debit cards and prepaid cards, the full authorization reversal service works for debit cards and prepaid cards in addition to credit cards.

> **Important**
>
> You cannot perform an authorization reversal if a transaction is in a review state, which can occur if you use a fraud management service. You must reject the

> transaction prior to authorization reversal. For more information, see the fraud management documentation in the Business Center.

Automatic Partial Authorization Reversals

Automatic partial authorization reversals are supported for:

- Credit cards
- Debit cards and prepaid cards.
- Quasi-cash.

If the capture amount is less than the authorization amount, Cybersource automatically performs a partial authorization reversal before it sends the capture request to the processor. The results of a successful partial authorization reversal are:

- The capture amount matches the new authorization amount at the payment card company.
- The hold on the unused credit card funds might be released. The issuing bank decides whether or not to release the hold on unused funds.
  Not all issuers act on a request for a partial authorization reversal. Therefore, Cybersource cannot guarantee that the funds will be released.

Captures

A capture is a follow-on transaction to an authorization. It is used to transfer the authorized funds from the customer's account to the merchant account. To link the authorization transaction to the capture transaction, you include a request ID in your capture request. This request ID is returned to you in the authorization response.
Captures are typically not performed in real time. They are placed in a batch file and sent to the processor, and the processor settles all of the captures at one time. In most cases, these batch files are sent and processed outside of the merchant's business hours. It usually takes 2 to 4 days for the acquiring financial institution to deposit the funds into the merchant account.
When fulfilling only part of a customer’s order, do not capture the full amount of the authorization. Capture only the cost of the delivered items. When you deliver the remaining items, request a new authorization, and then capture the new authorization.

> **Important**
>
> It is not possible to perform a capture if a transaction is in a review state, which can occur if you use a fraud management service. You must accept the transaction prior to capture. For more information, see the fraud management documentation in the Business Center.

Capture Workflow

The capture workflow begins when you send a request for a capture.

1. The merchant sends a request for a capture to Cybersource.

2. For online captures, Cybersource validates the order information then sends an online capture to the payment processor. For offline captures, Cybersource stores the capture request in a batch file and sends the batch file to the payment processor after midnight.
3. The processor validates the request and forwards it to the issuing bank.
4. The issuing bank transfers funds to the acquiring bank.

Important

The payment processor does not notify Cybersource that the money has been transferred. To ensure that all captures are processed correctly, you should reconcile your capture requests with the capture reports from your processor.

Credits

Credits are payment refunds from a merchant to the cardholder after a cardholder pays for a product or service and that payment is captured by the merchant. When a credit request is successful, the issuer transfers funds from the merchant bank (acquirer) account to the customer's account. It typically takes 2 to 4 days for the acquirer to transfer funds from your merchant account.

Warning

You should carefully control access to the credit service. Do not request this service directly from your customer interface. Instead, incorporate this service as part of your customer service process. This process reduces the potential for fraudulent transactions.

There are two basic types of credits: refunds and stand-alone credits.

Refunds

Refunds, also known as follow-on credits , use the capture request ID to link the refund to a specific transaction. This request ID is returned during the capture request (also known as a settlement) and is used in all subsequent refunds associated with the original capture. The request ID links the transaction to the customer’s billing and account information, so you are not required to include those fields in the credit request. **However, when you combine a request for a refund with a request for another service, such as the tax calculation service, you must provide the customer’s billing and account information.** Unless otherwise specified, refunds must be requested within 180 days of a settlement. You can request multiple refunds against a single To perform multiple refunds, use the same request ID in each request.

Stand-Alone Credits

Stand-alone credits are not connected to an original transaction. Stand-alone credits do not have a time restriction, and they can be used to issue refunds more than 180 days after a transaction settlement.

Credit Workflow

The credit workflow begins when you send a request for a credit.

A credit does not happen in real time. All of the credit requests for a day are typically placed in a file and sent to the processor as a single batch transaction. In most cases, the batch transaction is settled overnight.

1. The merchant sends a request for a credit to Cybersource.
2. For online credits, Cybersource validates the order information then sends an online credit to the payment processor. For offline credits, Cybersource stores the credit request in a batch file and sends the batch file to the payment processor after midnight.
3. The processor validates the request and forwards it to the acquiring bank.
4. The acquiring bank transfers funds to the issuing bank.

Voids

A void cancels a capture or credit request that was submitted but not yet processed by the processor.

Capture and credit requests are usually submitted once a day. A void request is declined when the capture or credit request has already been sent to the processor.

After a void is processed, you cannot credit or capture the funds. You must perform a new transaction to capture or credit the funds. Further, when you void a capture, a hold remains on the authorized funds. If you are not going to re-capture the authorization, and if your processor supports authorization reversal after void (ARAV), you should request an authorization reversal to release the hold on the unused funds.

A void uses the capture or credit request ID to link the transactions. The authorization request ID is used to look up the customer's billing and account information, so there is no need to include those fields in the void request. You cannot perform a follow-on credit against a capture that has been voided.

Payment Features

You can apply features to different payment services to enhance the customer payment processing experience. This section includes an overview of these features:

- *Card-Present Authorizations* on page 38
- *Debit and Prepaid Card Payments* on page 38
- *Airline Data* on page 39
- *Interchange Optimization* on page 45
- *Japanese Payment Options* on page 46
- *Level II and Level III Data* on page 47
- *Mastercard Bill Payments* on page 48
- *Mastercard Expert Monitoring Solutions* on page 48
- *Payer Authentication* on page 49
- *Processing Payments Using Credentials* on page 310

- *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 50
- *Split Shipments* on page 50
- *Token Management Service* on page 54

Card-Present Authorizations

For card-present transactions, the presence of the payment card is established during the authorization service. These are the basic types of card-present authorizations:

- EMV authorization: Authorization that is based on the EMV chip embedded in the cardholder's card.
- Magnetic stripe authorization: Authorization that is based on the magnetic stripe on the back of the cardholder's card.
- Hand-keyed authorization: Authorization that is based on you manually entering the card information into the payment terminal.
- Cash advance authorization: Authorization for withdrawing cash against a cardholder's credit card limit at their bank.
-

After you complete a card-present authorization, using these follow-on services enables you to complete the full payment workflow:

- Capture
- Contact EMV capture
- Stand-alone credit
- Authorization reversal
- Void

Related Information

- See the *Card Present Connect | Retail Integration Guide* for more information about the retail transactions.
- See *Card Present Connect | Retail Processing* on page 128 for information that shows you how to process retail payments.
- See the *Card Present Connect | Mass Transit Developer Guide* for more information about the mass transit transactions.
- See *Card Present Connect | Mass Transit Processing* on page 152 for information that shows you how to process mass transit payments.

Debit and Prepaid Card Payments

Debit cards are linked to a cardholder's checking account. A merchant who accepts the debit card can deduct funds directly from the linked cardholder's account.
You can process debit cards using these services:

- Credit card services
- PIN debit services

- Partial authorizations, which are a special feature available for debit cards
- Balance inquiries, which are a special feature available for debit cards

Requirements

In Canada, to process domestic debit transactions on Visa Platform Connect with Mastercard, you must contact customer support to have your account configured for this feature.

Related Information

- See *Standard Payment Processing* on page 58 for information that shows you how to use credit card services.
- See *Debit and Prepaid Card Processing* on page 202 for information that shows you how to process authorizations that use a debit or prepaid card.

Airline Data

Airline data processing goes beyond basic payment transactions by allowing you to process specific travel data. This requires you to submit additional information, such as:

- Carrier
- Departure Date
- Destination Airport
- Purchase Date
- Originating Airport
- Ticket Class
- Trip Legs

Supported Card Types

- American Express
- Discover
- Mastercard
- Visa

Supported Acquirers

These Visa Platform Connect acquirers are supported for airline data processing:

- Agricultural Bank of China (ABC)
- Ahli United Bank in Bahrain
- Arab African International Bank (AAIB)
- Asia Commercial Bank (ACB)
- Auckland Savings Bank (ASB)
- Axis Bank Ltd. of India
- Bangkok Bank Ltd.
- Bank Muscat of Oman

- Bank of Ayudhya (BAY)
- Bank of China (BOC)
- Bank of Communications
- Bank Sinarmas (Omise Ltd.)
- Banque Pour Le Commerce Exterieur Lao (BCEL)
- Barclays Bank Mauritius Ltd.
- Barclays Bank Botswana
- Barclays Bank of Ghana Ltd., Barclays Bank of Tanzania Ltd., and Barclays Bank of Uganda Ltd.
- Barclays Bank of Kenya
- Barclays Bank of Zambia
- Barclays Bank Seychelles
- BC Card Co., Ltd.
- BLOM Bank
- Cathay United Bank (CUB)
- Citibank Hongkong and Macau
- Citibank Singapore Ltd.
- Commercial Bank of Qatar
- CrediMax (Bahrain)
- CTBC Bank Ltd.
- FirstRand Bank
- Global Payments Asia Pacific
- Habib Bank Ltd. (HBL)
- HDFC Bank Ltd. of India
- I&M Bank
- ICICI of India
- Korea Exchange Bank (KEB)
- Mashreq
- National Bank of Abu Dhabi (NBAD)
- National Bank of Kuwait (NBK)
- National Commercial Bank
- Network International
- Overseas Chinese Banking Corp (OCBC)
- Promerica in Honduras and Nicaragua
- Qatar National Bank (QNB Group)
- Raiffeisenbank
- Rosbank
- Taishin Bank Ltd.
- United Overseas Bank (UOB) in Singapore and Vietnam
- United Overseas Bank (UOB) in Thailand
- Vietcombank
- VTB24

- Wing Lung Bank

Requirement

When you are ready to go live with airline data processing, contact Cybersource Customer Support to have your account configured to process airline data. If your account is not enabled, and you try to send airline transactions, you will receive an error for invalid data.

Related Information

- See *Airline Data Processing* on page 214 for information that shows you how to process payments that include airline data.

Cybersource Airline Data Processing

Cybersource does not store airline data. Instead, it functions as a pass-through service for the data. Cybersource enforces only the minimal level of field validation.

When you request an airline service, Cybersource responds with certain fields and values to indicate whether the airline data was processed. The response fields for each service are:

- Authorization: **processingInformation.enhancedDataEnabled**
- Capture: **processingInformation.enhancedDataEnabled**
- Credit: **processingInformation.enhancedDataEnabled**

The possible values for the response fields are:

- `Y`: the airline data was included in the request to the processor.
- `N`: the airline data was not included in the request to the processor.

Cybersource temporarily disables your account's airline data processing capability and contacts you if your airline data transactions produce batching errors when the information is sent to the processor. If this happens, your request is not rejected, but you receive one of the above listed fields with the `N` value in the response indicating that airline data in the request has been ignored and not sent to the processor.

Airline Data Reference Information

This section contains reference information that is useful when using Airline Data.

Airline Document Type Codes

To indicate the purpose of a purchase, set the **travelInformation.transit.airline.documentType** field to a value listed in the Code column.

Airline Document Type Codes

Code	Description
01	Passenger ticket
02	Additional collection
03	Excess baggage

Code	Description
04	Miscellaneous charge order (MCO) or prepaid ticket authorization
05	Special service ticket
06	Supported refund
07	Unsupported refund
08	Lost ticket application
09	Tour order voucher
10	Ticket by mail
11	Undercharge adjustment
12	Group ticket
13	Exchange adjustment
14	SPD or air freight
15	In-flight adjustment
16	Agency passenger ticket
17	Agency tour order or voucher
18	Agency miscellaneous charge order (MCO)
19	Agency exchange order
20	Agency group ticket
21	Debit adjustment for duplicate refund or use
22	In-flight merchandise order
23	Catalogue merchandise order
24	In-flight phone charges
25	Frequent flyer fee or purchase
26	Kennel charge
27	Animal transportation charge
28	Firearms case
29	Upgrade charge
30	Credit for unused transportation
31	Credit for class of service adjustment
32	Credit for denied boarding
33	Credit for miscellaneous refund

Code	Description
34	Credit for lost ticket refund
35	Credit for exchange refund
36	Credit for overcharge adjustment
37	Credit for multiple Unused tickets
38	Exchange order
39	Self-service ticket
41	In-flight duty-free purchase
42	Senior citizen discount booklets
43	Club membership fee
44	Coupon book
45	In-flight charges
46	Tour deposit
47	Frequent flyer overnight delivery charge
48	Frequent flyer fulfillment
49	Small package delivery
50	Vendor sale
51	Miscellaneous taxes or fees
52	Travel agency fee
60	Vendor refund or credit
64	Duty free sale
65	Preferred seat upgrade
66	Cabin upgrade
67	Lounge or club access or day pass
68	Agent assisted reservation or ticketing fee
69	Ticket change or cancel fee
70	Trip insurance
71	Unaccompanied minor
72	Standby fee
73	Curbside baggage
74	In-flight medical equipment

Code	Description
75	Ticket or pass print fee
76	Checked sporting or special equipment
77	Dry ice fee
78	Mail or postage fee
79	Club membership fee or temporary trial
80	Frequent flyer activation or reinstatement
81	Gift certificate
82	Onboard or in-flight prepaid voucher
83	Optional services fee
84	Advance purchase for excess baggage
85	Advance purchase for preferred seat upgrade
86	Advance purchase for cabin upgrade
87	Advance purchase for optional services
88	Wi-Fi
89	Packages
90	In-flight entertainment or internet access
91	Overweight bag fee
92	Sleep sets
93	Special purchase fee

Ancillary Service Category Codes

To indicate the service provided in an ancillary purchase, set the **travelInformation.transit.airline.ancillaryInformation.service[].categoryCode** and **travelInformation.transit.airline.ancillaryInformation.service[].subCategoryCode** fields to a value listed in the Ancillary Service Category Code column.

Ancillary Service Category Codes

Ancillary Service Category Codes	Description
BF	Bundled service
BG	Baggage fee
CF	Change fee
CG	Cargo

Ancillary Service Category Codes	Description
CO	Carbon offset
FF	Frequent flyer
GF	Gift card
GT	Ground transport
IE	In-flight entertainment
LG	Lounge
MD	Medical
ML	Meal or beverage
OT	Other
PA	Passenger assist fee
PT	Pets
SA	Seat fees
SB	Standby
SF	Service fee
ST	Store
TS	Travel service
UN	Unaccompanied travel
UP	Upgrades
WI	Wi-Fi

Interchange Optimization

Interchange fees are per-transaction transfer fees charged by your acquirer. The fee amount is based in part on the transaction amount that the acquirer submits to the payment network for clearing and settlement. Interchange optimization can help to reduce these fees for card-present transactions.

Payment Cards Supported with Interchange Optimization

- Mastercard
- Visa

Automatic Authorizations

Interchange optimization works by automatically performing additional authorization transactions for two types of card-not-present scenarios.

Automatic Authorization Refresh	**If a capture request occurs more than 6 days after the date of the original authorization, the processor automatically obtains a fresh authorization for the capture amount.**
Automatic Partial Authorization Reversal	**If the capture does not need a fresh authorization but the capture amount is less than the authorization amount, the processor automatically performs a partial authorization reversal. The reversal releases the hold on unused credit card funds and ensures that the settlement amount matches the authorization amount.**

How Interchange Optimization Transactions are Tracked

To find out when the processor performed automatic authorizations, see the daily processor report.

Limitations

- Interchange optimization does not apply to transactions in which the payment card is present at the merchant's physical place of business.
- Interchange optimization is not supported with incremental authorizations.

Requirement

Contact customer support to enable interchange optimization for your account.

Related Information

Merchant Financial Institutions (Acquirers) **on page 19**

Japanese Payment Options

Japanese payment options (JPO) extend the Cybersource payment card processing features to support payment methods used only in Japan. Japanese issuers, cardholders, merchants, and acquirers recognize payment methods that clarify the nature of a payment. JPO provides for more fine-grained identification of one-time payments and installment payments. You can offer your customers JPO payment methods that they select at the time of purchase.
JPO supports these payment methods:

- Single payment
- Bonus payment
- Installment payment
- Revolving payment
- Combination of bonus payment and installment payment

> **Important**
>
> Requests with Japanese payment options are accepted independently of your agreements with acquirers. When you submit a request with one of these payment options but do not have the necessary contracts and agreements in place, an error might not occur until the acquirer processes the settlement file.

For more information about the Japanese payment options, contact Customer Support of Cybersource KK (Japan).

Payment Cards Supported with JPO

JPO is supported for the Sumitomo Mitsui Card Co. acquirer with transactions that use Visa payment cards issued in Japan.

Services Supported with JPO

Authorization service.

Requirements

- You have signed a contract with your acquirer.
- You have contacted your account provider for details about contracts and funding cycles. The funding cycle could differ when using JPO.
- Card holders who want to use JPO have signed a contract with an issuing bank.
- You have confirmed payment option availability with your account provider and card holder before implementing one of these payment options.

Related Information

- See *Japanese Payment Options Processing* on page 232 for information that shows you how to process payments using JPO.

Level II and Level III Data

For business to business customers, Level II and Level III processing can provide lower interchange rates in exchange for providing more information during a transaction. Support for Level II and Level III data processing is processor and card specific.

Level II Data

Level II cards, which are also called Type II cards, provide customers with additional information on their credit card statements about their purchases. Level II cards enable customers to easily track the amount of sales tax they pay and to reconcile transactions with a unique customer code. There are two categories of Level II cards:

- Business/corporate cards are given by businesses to employees for business-related expenses such as travel and entertainment or for corporate supplies and services.

- Purchase/procurement cards are used by businesses for expenses such as supplies and services. These cards are often used as replacements for purchase orders.

Level III Data

You can provide Level III data for purchase/procurement cards, which are used by businesses for expenses such as supplies and services. These cards are often used as replacements for purchase orders. The Level III data is forwarded to the company that made the purchase. It enables the company to manage its purchasing activities.

Related Information

- See *Level II Processing* on page 251 for information that shows you how to process transactions that include Level II data.
- See *Level III Processing* on page 261 for information that shows how to process transactions that include Level III data.

Mastercard Bill Payments

In Brazil, you can participate in a Mastercard Bill Payment program. If your account is enrolled in the program, your customers can use their Mastercard payment cards to make payments on their outstanding bills.

> **Important**
>
> A Mastercard card payment at the point of sale (POS) when goods or services are purchased is not part of the Mastercard Bill Payment program.

When you send an authorization request for a Mastercard Bill Payment, include the API field that specifies the bill payment type.

Limitation

The Mastercard Bill Payment program supports only bills paid in Brazil using Mastercard payments cards with Visa Platform Connect.

Requirements

Sign up with Mastercard to participate in their bill payment program.

Related Information

- See *Mastercard Bill Payment Processing* on page 274 for information that shows you how to process Mastercard Bill Payments.

Mastercard Expert Monitoring Solutions

Mastercard Expert Monitoring Solutions provides a predictive, behavior-based fraud score in real time during authorizations for card-not-present transactions. The score

indicates the likelihood that the requested transaction is fraudulent and the type of fraud that is suspected.
To assign the fraud score for a transaction, Mastercard compares the customer's transaction data to their transaction behavior history and to a regional card-not-present fraud detection model. The resulting score is returned in the body of the response message.

Limitations

This feature is supported on Mastercard Payment cards issued in the US only.
This feature is supported with Visa Platform Connect only.

Requirement

Contact customer support to enable Mastercard Expert Monitoring Solutions for your account.

> **Important**
>
> After this feature is enabled for your account, Mastercard returns a fraud score for all your card-not-present authorization requests for Mastercard payment cards issued in the US.

Related Information

- See *Mastercard Expert Monitoring Solutions Processing* on page 277 for information that shows you how to obtain the transaction fraud score determined by Mastercard Expert Monitoring Solutions.

Payer Authentication

Payer authentication is run before a transaction is submitted for authorization. Most of the time payer authentication is bundled with authorization so that after payer authentication happens, the transaction is automatically submitted for authorization. Payer authentication and authorization can be configured to occur as separate operations. This section shows you how to run payer authentication as a separate process and pass the payer authentication data when seeking authorization for a transaction.
Payer authentication consists of a two-step verification process that adds an extra layer of fraud protection during the payment process. During transactions, the transaction device, location, past purchasing habits, and other factors are analyzed for indications of fraud. This process collects customer data during the transaction from at least two of these three categories:

- Something you have: A payment card or a payment card number
- Something you know: A password or pin
- Something you are: Facial recognition or fingerprint

Each of these payment card companies has its own payer authentication product:

- American Express: SafeKey

- Discover: ProtectBuy
- JCB: J/Secure
- Mastercard: Identity Check
- Visa: Visa Secure

Payer authentication can be used to satisfy the Strong Customer Authentication (SCA) requirement of the Payment Services Directive (PSD2). SCA applies to the European Economic Area (EEA) and the United Kingdom. SCA requires banks to perform additional checks when customers make payments to confirm their identity.

Related Information

- See the *Payer Authentication Developer Guide* for more information about payer authentication.
- See *Payer Authentication Processing* on page 281 for information about how to process payments with payer authentication.

Relaxed Requirements for Address Data and Expiration Date in Payment Transactions

With relaxed requirements for address data and the expiration date, not all standard payment request fields are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required.

Related Information

- See *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300 for information about how to process payments with relaxed requirements for address data and expiration date.

Split Shipments

Split shipments enable you to split an order into multiple shipments with multiple captures. You can use this feature when a customer orders a product that is not yet available, or when one or some products are available but not all. You are able to request multiple partial captures for one authorization, multiple authorizations and one capture, or an authorization and a sale.
Cybersource provides the split shipment services for authorizations and captures. There are three scenarios and actions you can take:

- Multiple authorizations—Request more than one authorizations; when the order is placed for the unavailable product and after the product becomes available to ship.
- Multiple partial captures—Request an authorization, and then request multiple partial captures for the amount of the products you ship. When the remaining product becomes available, ship it and request another capture.
- Multiple authorizations with multiple partial captures—Request more than one authorizations and captures when all the products in the order are not available for

immediate shipment. After the other products become available, request another authorization, and then a capture when you ship the remaining product.

How Split Shipments Transactions are Linked

All transactions for a split shipment are linked together in the Business Center and in reports. When you split an order into multiple shipments with multiple partial captures, Cybersource requests the additional authorizations for you.

Obtaining the Status of a System-Generated Authorization

> **Important**
>
> A system-generated authorization is not performed in real time. The response message that you receive indicates that the request was received, not whether it was approved or declined.

A system-generated authorization can be declined for the same reasons that a regular authorization can be declined. Cybersource recommends you use one of following methods to obtain the status of the system-generated authorization request before shipping the product:

- Business Center—Use the capture request ID to search for the follow-on capture. The details for all related transactions are displayed on the Transaction Details page. It can take a maximum of 6 hours for the status of the system-generated authorization request to be available.
- Transaction Detail API—You must use version 1.3 or later of the report and include the parameter **includeExtendedDetail** in your query. It can take a maximum of 6 hours for the status of the system-generated authorization request to be available.
- Transaction Exception Detail Report—Cybersource recommends you use this report on a daily basis to identify transactions that were declined.

Additional Authorizations

When you need an additional authorization for an order, you can use the **link-to-request** field to link follow-on authorizations to the original authorization in addition to the basic fields required for every authorization request. The follow-on authorization is linked to the original authorization in the Business Center and in reports. The captures for these authorizations are also linked to the original authorization in the Business Center and in reports.

For an additional authorization on a processor that supports merchant-initiated transactions, the authorization request must include the subsequent authorization fields that are required for merchant-initiated transactions.

Additional Captures

When you need an additional capture for an order, Cybersource performs a system-generated authorization for additional capture requests using the payment data from the original authorization. The system-generated authorization is linked to the original authorization in the Business Center and in reports. The captures are linked to the authorizations in the Business Center and in reports through the request IDs as with any capture.

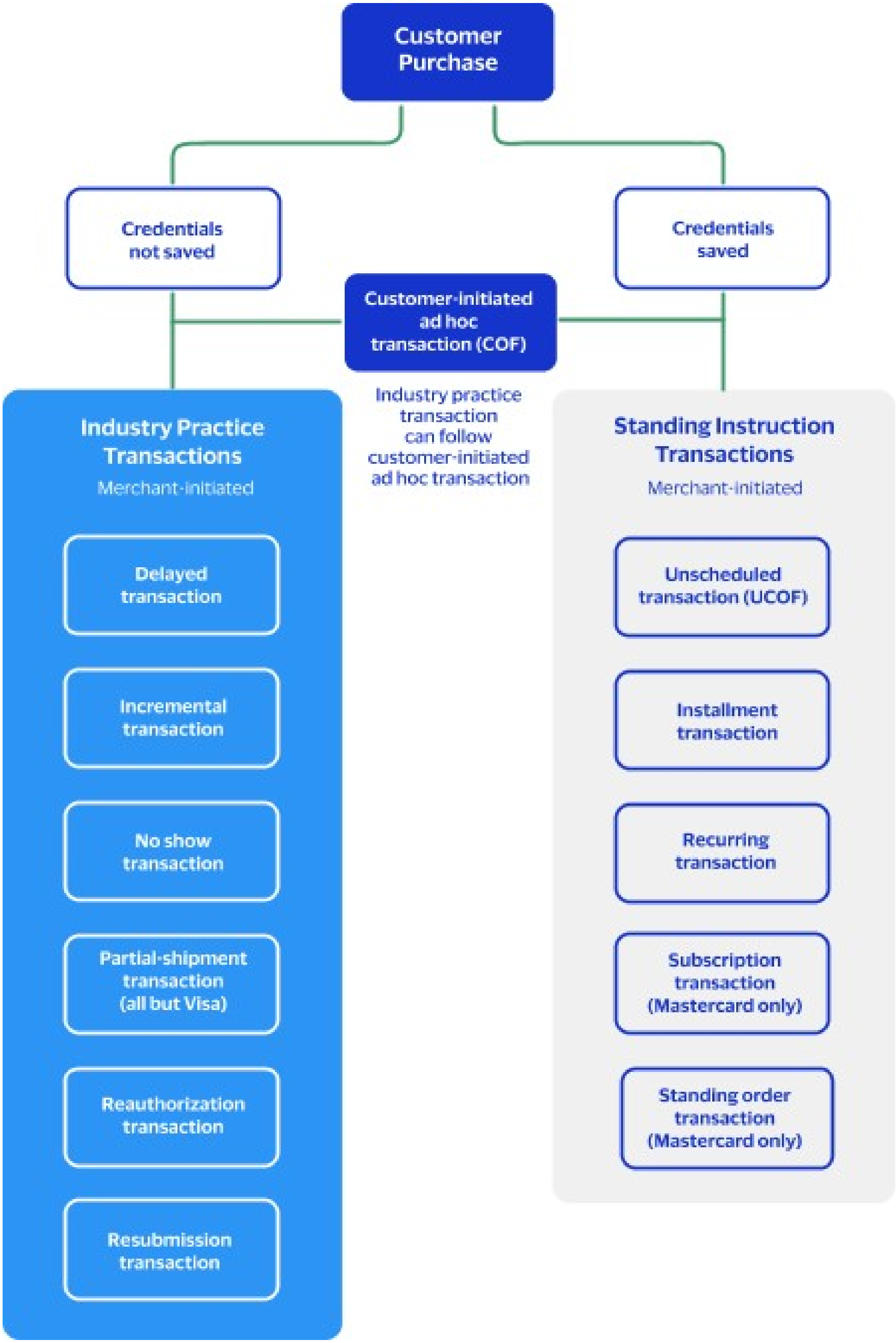
Related Information

- See *Authorizing a Sale for a Product Not Yet Available* on page 303 for guidelines on how to process a payment when a product is not available.
- See *Processing an Authorization and Two Captures for Multiple Products* on page 307 or *Processing Two Authorizations and a Capture for Multiple Products* on page 305 for guidelines on how to process payments for multiple products.

Introduction to Credentialed Transactions

Credentialed transactions are transactions that involve either storing a customer's payment credentials for future transactions or using a customer's already stored payment credentials. When processing a credentialed transaction, you must indicate the type of credentialed transaction and the reason for the transaction. Credentialed transactions are also known as credential-on-file (COF) transactions.
There are several types of credentialed transactions:

- Customer-Initiated Transactions (CITs): Any transaction a customer is actively participating in such as making a card-present payment, completing an online checkout, or by using a stored credential. CIT transactions can store the customer's credentials in your system for future CITs or merchant-initiated transactions.
- Merchant-Initiated Transactions (MITs): Any transaction a merchant initiates without the customer's participation such as an industry practice transaction or a standing instruction transaction.
  - Industry Practice Transactions: MITs that are performed as subsequent transactions to a CIT because the initial transaction could not be completed in one transaction. Not every industry practice transaction involves a stored credential. If a stored credential is used only for one transaction, that transaction is not considered a credentialed transaction.
  - Standing Instruction Transactions: MITs that are performed to follow agreed-upon instructions from the customer for the provision of goods and services.


Customer Purchase
Credentials not saved
Credentials saved
Customer-initiated ad hoc transaction (COF)
Industry practice transaction can follow customer-initiated ad hoc transaction
Industry Practice Transactions
Merchant-initiated
Delayed transaction
Incremental transaction
No show transaction
Partial-shipment transaction (all but Visa)
Reauthorization transaction
Resubmission transaction
Standing Instruction Transactions
Merchant-initiated
Unscheduled transaction (UCOF)
Installment transaction
Recurring transaction
Subscription transaction (Mastercard only)
Standing order transaction (Mastercard only)

MIT Types

Supported Services

These are the supported merchant-initiated services:

- Delayed Authorization
- Incremental Transactions
- Installment Transactions
- Mastercard Standing Order Transactions
- Mastercard Subscription Transactions
- No-Show Transactions
- Reauthorization
- Recurring Transactions
- Resubmission
- Unscheduled Credentials-on-File Transactions

The service determines the reason for the credentialed transaction.

Token Management Service

The Token Management Service (TMS) enables you to replace personally identifiable information (PII), such as the primary account numbers (PANs), with unique tokens. These tokens do not include the PII data, but act as a placeholder for the personal information that would otherwise need to be shared. By using tokens, businesses can provide a secure payment experience, reduce the risk of fraud, and comply with industry consumer security regulations such as PCI-DSS.

TMS links tokens across service providers, payment types, and channels for sellers, acquirers, and technology partners. TMS tokenizes, securely stores, and manages the primary account number (PAN), the payment card expiration date, electronic check details, and customer data. TMS also enables you to create a network token of a customer's payment card.

> **Important**
>
> Due to mandates from the Reserve Bank of India, Indian merchants cannot store PANs. Use network tokenization instead.

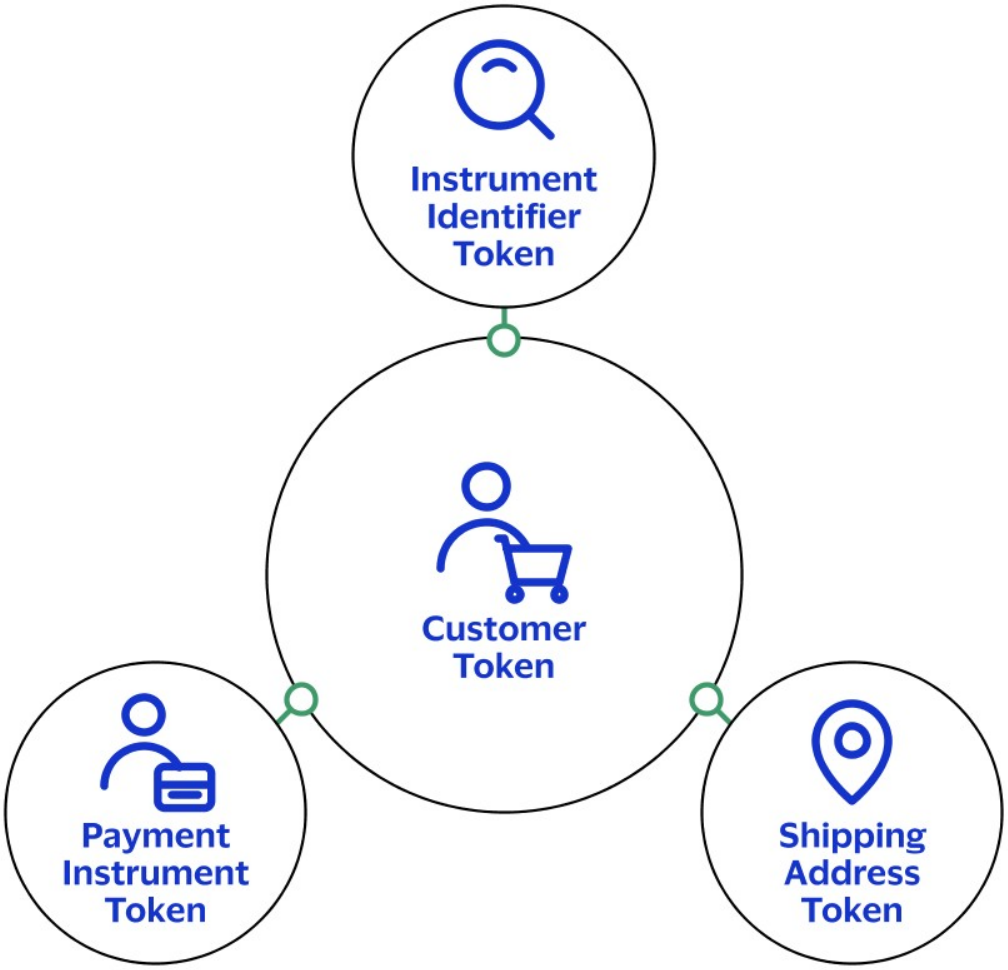
You can manage sensitive data securely by creating, retrieving, updating, and deleting tokens through the *TMS API*.

TMS simplifies your PCI DSS compliance. TMS passes tokens back to you that represent this data. You then store these tokens in your environment and databases instead of storing customer payment details.

TMS protects sensitive payment information through tokenization and secures and manages customer data using these token types:

- Customer tokens
- Instrument identifier tokens
- Payment instrument tokens
- Shipping address tokens

These TMS tokens can be used individually, or they can be associated with one customer token:



TMS Token Types

Related Information

- See the *Token Management Service Developer Guide* for more information about the TMS.
- See *Token Management Service Processing* on page 479 for information that shows you how to process payments using the TMS.

Testing the Payment Services

To ensure that requests are processed correctly, you must test the basic success and error conditions for each service you plan to use.

Requirements for Testing

Before you can test, contact customer support to activate the credit card services and configure your account for testing. You must also contact your processor to set up your processor account.

> **Important**
>
> When building your connection to the Cybersource payment gateway, ensure that you have implemented controls to prevent card testing or card enumeration attacks on your platform. For more information, see the *best practices guide*.
> When we detect suspicious transaction activity associated with your merchant ID, including a card testing or card enumeration attack, Cybersource reserves the right to enable fraud management tools on your behalf in order to mitigate the attack. The fraud team might also implement internal controls to mitigate attack activity. These controls block traffic that is perceived as fraudulent. Additionally, if you are using one of our fraud tools and experience a significant attack, our internal team might modify or add rules to your configuration to help prevent the attack and minimize the threat to our infrastructure. However, any actions taken by Cybersource would not replace the need for you to follow industry standard best practices to protect your systems, servers, and platforms.

Follow these requirements when you test your system:

- Use your regular merchant ID.
- Use a real combination for the city, state, and postal code.
- Use a real combination for the area code and telephone number.
- Use a nonexistent account and domain name for the customer's email address.
- REST API test endpoint: `POST https://apitest.cybersource.com/pts/v2/payments`

Test Card Numbers

Use these payment card numbers to test the authorization, capture, and credit services. Remove the spaces from the test card numbers when sending them to the test system. Do not use real payment card numbers. To test card types that are not included in the list, use an account number that is in the card's BIN range. For best results, try each test with a different service request and with different test payment card numbers.

> **Important**
>
> The test card numbers that are provided are formatted with Xs for zeroes in the card number. When testing with the card numbers, replace the Xs with a 0 (zero).

- American Express—3782 8224 631X XX5
- Discover—6X11 1111 1111 1117
- JCB—3566 1111 1111 1113
- Maestro (International)

  - 5X33 9619 89X9 17
  - 5868 2416 0825 5333 38
- Maestro (UK Domestic)—the issue number is not required for Maestro (UK Domestic) transactions.
  - 6759 4111 XXXX XXX8
  - 6759 56XX 45XX 5727 054
  - 5641 8211 1116 6669
- Mastercard
  - 2222 42XX XXXX 1113
  - 2222 63XX XXXX 1125
  - 5555 5555 5555 4444
- UATP—1354 1234 5678 911
- Visa—4111 1111 1111 1111

Using Amounts to Simulate Errors

You can simulate error messages by requesting authorization, capture, or credit services with specific amounts that trigger the error messages. These triggers work only on the test server, not on the production server.
Each payment processor uses its own error messages. For more information, see: *REST API Testing Guide* .

Test American Express Card Verification

Before using CVN with American Express, it is strongly recommended that you follow these steps:

1. Contact customer support to have your account configured for CVN. Until you do this, you will receive a `1` in the **processorInformation.cardVerification.resultCode** response field.
2. Test your system in production using a small currency amount, such as one currency unit. Instead of using the test account numbers, use a real payment card account number, and send an incorrect CVN in the request for authorization. The card should be refused and the request declined.

Standard Payment Processing

This section shows you how to process various authorization, capture, credit, and sales transactions.

Additional Resources for Processing Payments

For more information, see these guides:

- *API field reference guide for the REST API*
- Github repositories: *https://github.com/Cybersource*

Basic Authorizations

This section provides the information you need in order to process a basic authorization.

Supported Card Types

All supported card types can process basic authorizations. For a list of all supported card types, see *Payment Processors* on page 20.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Declined Authorizations

If an authorization is declined, you can use response categories to help you decide whether to retry or block a declined transaction. These response fields provide additional information:

- **paymentInsightsInformation.responseInsights.category**
- **paymentInsightsInformation.responseInsights.categoryCode**

Category codes have possible values (such as `01`) each of which corresponds to a category that contains a description.
You cannot retry this category code and category:

- `01 ISSUER_WILL_NEVER_APPROVE`

For these values, you can retry the transaction a maximum of 15 times over a period of 30 days:

- `02 ISSUER_CANNOT_APPROVE_AT_THIS_TIME`
- `03 ISSUER_CANNOT_APPROVE_WITH_THESE_DETAILS`: Data quality issue. Revalidate data prior to retrying the transaction.
- `04 GENERIC_ERROR`
- `97 PAYMENT_INSIGHTS_INTERNAL_ERROR`
- `98 OTHERS`
- `99 PAYMENT_INSIGHTS_RESPONSE_CATEGORY_MATCH_NOT_FOUND`

Required Fields for Processing a Basic Authorization

Use these required fields for processing a basic authorization.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

Related Information

- *API field reference guide for the REST API*

Country-Specific Required Fields for Processing a Basic Authorization

Use these country-specific required fields to process a basic authorization.

Argentina

merchantInformation.taxId	**Required for Mastercard transactions.**
merchantInformation.transactionLocalDateTime	**Required in Argentina when the time zone is not included in your account. Otherwise, this field is optional.**

Brazil

paymentInformation.card.sourceAccountType	**Required for combo card transactions.**
paymentInformation.card.sourceAccountTypeDetails	**Required for combo card line-of-credit and prepaid-card transactions.**

Chile

merchantInformation.taxId	**Required for Mastercard transactions.**

Egypt

paymentInformation.card.cardType	**Required for Meeza transactions. Set to `067`.**
merchantInformation.merchantDescriptor.country	**Required for Meeza transactions. Set to `EG`.**

Paraguay

merchantInformation.taxId	**Required for Mastercard transactions.**

Saudi Arabia

processingInformation.authorizationOptions.transactionMode	**Required only for merchants in Saudi Arabia.**

Taiwan

paymentInformation.card.hashedNumber	**Required only for merchants in Taiwan.**

Related Information

- *API field reference guide for the REST API*

REST Interactive Example: Processing a Basic Authorization

```
Simple Authorization(Internet)
```

Live Console URL: *https://developer.cybersource.com/api-reference-assets/index.html#payments_payments_process-a-payment*

REST Example: Processing a Basic Authorization

Request

```
{
  "orderInformation": {
    "billTo": {
      "country": "US",
      "lastName": "Kim",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "firstName": "Kyong-Jin",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "usd"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111111111111111",
      "expirationMonth": "12",
      "type": "001"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links" : {
    "authReversal" : {
      "method" : "POST",
      "href" : "/pts/v2/payments/6461731521426399003473/reversals"
    },
    "self" : {
      "method" : "GET",
      "href" : "/pts/v2/payments/6461731521426399003473"
    },
    "capture" : {
      "method" : "POST",
      "href" : "/pts/v2/payments/6461731521426399003473/captures"
    }
  },
  "clientReferenceInformation" : {
    "code" : "1646173152047"
  },
  "id" : "6461731521426399003473",
  "orderInformation" : {
    "amountDetails" : {
      "authorizedAmount" : "100.00",
      "currency" : "usd"
    }
  },
  "paymentAccountInformation" : {
    "card" : {
      "type" : "001"
    }
  },
  "paymentInformation" : {
    "tokenizedCard" : {
      "type" : "001"
    },
    "card" : {
      "type" : "001"
    }
  },
 "paymentInsightsInformation" : {
    "responseInsights" : {
      "categoryCode" : "01"
    }
  },
  "processorInformation" : {
    "systemTraceAuditNumber" : "862481",
    "approvalCode" : "831000",
    "merchantAdvice" : {
      "code" : "01",
      "codeRaw" : "M001"
    },
    "responseDetails" : "ABC",
    "networkTransactionId" : "016153570198200",
    "consumerAuthenticationResponse" : {
      "code" : "2",
      "codeRaw" : "2"
    },
    "transactionId" : "016153570198200",
    "responseCode" : "00",
    "avs" : {
      "code" : "Y",
      "codeRaw" : "Y"
    }
  },
  "reconciliationId" : "6461731521426399003473",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2022-03-01T22:19:12Z"
}
```

Response to a Declined Request

```
{
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "errorInformation": {
    "reason": "PROCESSOR_ERROR",
    "message": "Invalid account"
  },
  "id": "6583553837826789303954",
  "paymentInsightsInformation": {
    "responseInsights": {
      "categoryCode": "01",
      "category": "ISSUER_WILL_NEVER_APPROVE"
    }
  },
  "pointOfSaleInformation": {
    "amexCapnData": "1009S0600100"
  },
  "processorInformation": {
    "systemTraceAuditNumber": "004544",
    "merchantNumber": "1231231222",
    "networkTransactionId": "431736869536459",
    "transactionId": "431736869536459",
   "responseCode": "111",
    "avs": {
      "code": "Y",
      "codeRaw": "Y"
    }
  },
  "status": "DECLINED"
}
```

Authorizations with Line Items

This section shows you how to process an authorization with line items.

The main difference between a basic authorization and an authorization that includes line items is that the **orderInformation.amountDetails.totalAmount** field, which is included in a basic authorization, is substituted with one or more line items that are included in a **lineItem[]** array.

Fields Specific to this Use Case

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

orderInformation.lineItems[].unitPrice	
orderInformation.lineItems[].quantity	
orderInformation.lineItems[].productCode	
orderInformation.lineItems[].productSku	Optional when **item_#_productCode** is set to `default`, `shipping_only`, `handling_only`, or `shipping_and_handling`
orderInformation.lineItems[].productName	Optional when **item_#_productCode** is set to `default`, `shipping_only`, `handling_only`, or `shipping_and_handling`

At a minimum, you must include the **orderInformation.lineItems[].unitPrice** field in order to include a line item in an authorization. When this field is the only field included in the authorization, the system sets:

- **orderInformation.lineItems[].productCode**: `default`
- **orderInformation.lineItems[].quantity**: `1`

For example, these three line items are valid.

```
"orderInformation": {
 "lineItems": [
  {
   "unitPrice": "10.00"
  },
  {
   "unitPrice": "5.99",
   "quantity": "3",
   "productCode": "shipping_only"
  },
  {
   "unitPrice": "29.99",
   "quantity": "3",
   "productCode": "electronic_good",
   "productSku": "12384569",
   "productName": "receiver"
  }
 ]
}
```

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing an Authorization with Line Items

Use these required fields for processing an authorization that includes line items.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

Related Information

- *API field reference guide for the REST API*

Country-Specific Required Fields for Processing an Authorization with Line Items

Use these country-specific required fields to process a process an authorization with line items.

Argentina

merchantInformation.taxId	**Required for Mastercard transactions.**
merchantInformation.transactionLocalDateTime	**Required in Argentina when the time zone is not included in your account. Otherwise, this field is optional.**

Brazil

paymentInformation.card.sourceAccountType	**Required for combo card transactions.**
paymentInformation.card.sourceAccountTypeDetails	**Required for combo card line-of-credit and prepaid-card transactions.**

Chile

merchantInformation.taxId	**Required for Mastercard transactions.**

Paraguay

merchantInformation.taxId	**Required for Mastercard transactions.**

Saudi Arabia

processingInformation.authorizationOptions.transactionMode	**Required only for merchants in Saudi Arabia.**

Related Information

- *API field reference guide for the REST API*

REST Example: Processing an Authorization with Line Items

Request

```
{
  "currencyConversion": {
   "indicator": "Y"
  },
  "paymentInformation": {
   "card": {
    "number": "4111111111111111",
    "expirationMonth": "12",
    "expirationYear": "2031"
   }
  },
  "orderInformation": {
   "amountDetails": {
    "currency": "USD",
    "exchangeRate": ".91",
    "originalAmount": "107.33",
    "originalCurrency": "eur"
   },
   "billTo": {
    "firstName": "John",
    "lastName": "Doe",
    "address1": "1 Market St",
    "locality": "san francisco",
    "administrativeArea": "CA",
    "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com"
  },
  "lineItems": [
   {
    "unitPrice": "10.00"
   },
   {
    "unitPrice": "5.99",
    "quantity": "3",
    "productCode": "shipping_only"
   },
   {
    "unitPrice": "29.99",
    "quantity": "3",
    "productCode": "electronic_good",
    "productSku": "12384569",
    "productName": "receiver"
   }
  ]
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6482385519226028804003/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6482385519226028804003"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6482385519226028804003/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1648238551902"
 },
 "id": "6482385519226028804003",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "117.94",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
   "tokenizedCard": {
    "type": "001"
   },
   "card": {
    "type": "001"
   }
  },
  "processorInformation": {
   "systemTraceAuditNumber": "191521",
   "approvalCode": "831000",
   "merchantAdvice": {
    "code": "01",
    "codeRaw": "M001"
   },
   "responseDetails": "ABC",
   "networkTransactionId": "016153570198200",
   "consumerAuthenticationResponse": {
    "code": "2",
    "codeRaw": "2"
   },
   "transactionId": "016153570198200",
   "responseCode": "00",
   "avs": {
    "code": "Y",
    "codeRaw": "Y"
   }
  },
  "reconciliationId": "64823855192260288040003",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-03-25T20:02:32Z"
 }
```

Authorizations with Payment Network Tokens

This section shows you how to successfully process an authorization with payment network tokens.

> **Important**
>
> Due to mandates from the Reserve Bank of India, Indian merchants cannot store personal account numbers (PAN). Use network tokens instead. For more information on network tokens, see *Network Tokenization* in the Token Management Service Developer Guide.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizations with Payment Network Tokens

Use these required fields for processing an authorization with payment network tokens.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

paymentinformation.tokenizedCard.cryptogram

paymentinformation.tokenizedCard.expirationMonth

paymentinformation.tokenizedCard.expirationYear

paymentInformation.tokenizedCard.transactionType

Related Information

- *API field reference guide for the REST API*

Optional Fields for Authorizations with Payment Network Tokens

You can use these optional fields to include additional information when processing an authorization with a payment network token.

clientReferenceInformation.code	
consumerAuthenticationInformation.cavv	**For 3-D Secure in-app transactions for Visa and JCB, set this field to the 3-D Secure cryptogram. Otherwise, set to the network token cryptogram.**
consumerAuthenticationInformation.ucafAuthenticationData	**For Mastercard requests using 3-D Secure, set this field to the Identity Check cryptogram.**
consumerAuthenticationInformation.ucafCollectionIndicator	**For Mastercard requests using 3-D Secure, set the value to 2.**

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	**Required only for transactions in the US and Canada.**
orderInformation.billTo.administrativeArea	**Required only for transactions in the US and Canada.**
processingInformation.commerceIndicator	
paymentInformation.tokenizedCard.cardType	**It is strongly recommended that you send the card type even if it is optional for your processor. Omitting the card type can cause the transaction to be processed with the wrong card type.**
paymentInformation.tokenizedCard.cryptogram	
paymentInformation.tokenizedCard.expirationMonth	**Set to the token expiration month that you received from the token service provider.**
paymentInformation.tokenizedCard.expirationYear	**Set to the token expiration year that you received from the token service provider.**
paymentInformation.tokenizedCard.number	**Set to the token value that you received from the token service provider.**
paymentInformation.tokenizedCard.requestorId	**Required on Visa Platform Connect**
paymentInformation.tokenizedCard.transactionType	

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizations with Payment Network Tokens

Request

```
{
  "orderInformation" : {
   "billTo": {
      "country": "US",
      "lastName": "Kim",
    "address1": "201 S. Division St.",
    "postalCode": "48104-2201",
    "locality": "Ann Arbor",
    "administrativeArea": "MI",
    "firstName": "Kyong-Jin",
    "email": "test@cybs.com"
    },
  "amountDetails" : {
   "totalAmount" : "100",
   "currency" : "USD"
  }
 },
  "paymentInformation" : {
  "tokenizedCard" : {
   "expirationYear" : "2031",
   "number" : "4111111111111111",
   "expirationMonth" : "12",
   "transactionType" : "1",
   "cryptogram" : "qE5juRwDzAUFBAkEHuWW9PiBkWv="
  }
 }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6838294805206235603954/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6838294805206235603954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6838294805206235603954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1683829480593"
  },
  "id": "6838294805206235603954",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "1"
    }
  },
  "reconciliationId": "60332034UHI9PRJ0",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-05-11T18:24:40Z"
}
```

Authorizations with a Card Verification Number

This section shows you how to process an authorization with a Card Verification Number (CVN).

CVN Results

The response includes a raw response code and a mapped response code:

- The raw response code is the value returned by the processor. This value is returned in the **processorInformation.cardVerification.resultCodeRaw** field. Use this value only for debugging purposes; do not use it to determine the card verification response.
- The mapped response code is the pre-defined value that corresponds to the raw response code. This value is returned in the **processorInformation.cardVerification.resultCode** field.

Even when the CVN does not match the expected value, the issuing bank might still authorize the transaction. You will receive a CVN decline, but you can still capture the transaction because it has been authorized by the bank. However, you must review the order to ensure that it is legitimate.

Settling authorizations that fail the CVN check might have an impact on the fees charged by your bank. Contact your bank for details about how card verification management might affect your discount rate.

When a CVN decline is received for the authorization in a sale request, the capture request is not processed unless you set the **processingInformation.authorizationOptions.ignoreCvResult** field to `true`.

CVN Results for American Express

A value of `1` in the **processorInformation.cardVerification.resultCode** field indicates that your account is not configured to use card verification. Contact customer support to have your account enabled for this feature.

CVN Results for Discover

When the CVN does not match, Discover refuses the card and the request is declined. The reply message does not include the **processorInformation.cardVerification.resultCode** field, which indicates that the CVN failed.

CVN Results for Visa and Mastercard

A CVN code of `D` or `N` causes the request to be declined with a reason code value of `230`. You can still capture the transaction, but you must review the order to ensure that it is legitimate.

Cybersource, not the issuer, assigns the CVN decline to the authorization. You can capture any authorization that has a valid authorization code from the issuer, even when the request receives a CVN decline.

When the issuer does not authorize the transaction and the CVN does not match, the request is declined because the card is refused. You cannot capture the transaction.

Fields Specific to this Use Case

Include this field with a standard authorization request when processing an authorization with a CVN:

- **paymentInformation.card.securityCode**

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing an Authorization with a Card Verification Number

Use these required fields for processing an authorization that includes a Card Verification Number (CVN).

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

paymentInformation.card.securityCode

paymentInformation.card.type

paymentInformation.card.securityCode

Related Information

- *API field reference guide for the REST API*

Optional Fields for Processing an Authorization with a Card Verification Number

You can use these optional fields to include additional information when processing an authorization with a card verification number.

paymentInformation.card.securityCodeIndicator

processingInformation.authorizationOptions.ignoreCvResult

REST Example: Processing an Authorization with a Card Verification Number

Request

```
{
  "paymentInformation": {
    "card": {
    "number": "4111111111111111",
    "expirationMonth": "12",
    "expirationYear": "2031",
    "type": "001",
    "securityCode": "999"
    }
  },
  "orderInformation": {
    "amountDetails": {
    "totalAmount": "49.95",
    "currency": "USD"
  },
   "billTo": {
    "firstName": "John",
    "lastName": "Doe",
    "address1": "1295 Charleston Rd.",
    "locality": "Mountain View",
    "administrativeArea": "CA",
    "postalCode": "94043",
    "country": "US",
    "email": "jdoe@example.com",
    "phoneNumber": "650-965-6000"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6554147587216874903954/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6554147587216874903954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6554147587216874903954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1655414758839"
  },
  "id": "6554147587216874903954",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "49.95",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "67546603C43Z6JWN",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-06-16T21:25:58Z"
}
```

Marketplace Authorizations with Foreign Retailers

Visa Platform Connect requires marketplaces to identify foreign retail transactions when the marketplace and issuer are in the European Economic Area (EEA), the U.K., and Gibraltar and the retailer is in a different country. For marketplace transactions, the marketplace is the merchant and the retailer is the sub-merchant. Marketplace foreign retail transactions are identified in the Business Center on the transactions details page.

Important

This feature is intended for captures. You can include this information in an authorization, but this is not the preferred method. The capture request data overrides the authorization request data.

Fields Specific to this Use Case

These fields are required for this use case:

aggregatorInformation.subMerchant.country	Set this value to the retailer country.
merchantInformation.merchantDescriptor.country	Set this value to the marketplace country.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing a Marketplace Authorization with a Foreign Retailer

Use these required fields for processing a basic authorization.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

aggregatorInformation.subMerchant.country	**Set this value to the retailer country.**
merchantInformation.merchantDescriptor.country	**Set this value to the marketplace country.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

Related Information

- *API field reference guide for the REST API*

REST Example: Processing an Marketplace Authorization with a Foreign Retailer

Request

```
{
   "aggregatorInformation" : {
      "subMerchant" : {
         "country" : "AU"
      }
   },
   {
   "orderInformation": {
      "billTo": {
         "country": "US",
         "lastName": "Kim",
         "address1": "201 S. Division St.",
         "postalCode": "48104-2201",
         "locality": "Ann Arbor",
         "administrativeArea": "MI",
         "firstName": "Kyong-Jin",
         "email": "test@cybs.com"
      },
      "amountDetails": {
         "totalAmount": "100.00",
         "currency": "GBP"
      }
   },
   {
   "merchantInformation" : {
      "merchantDescriptor" : {
         "country" : "GB"
      }
   },
   {
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "number": "4111111111111111",
         "expirationMonth": "12",
         "type": "001"
      }
   }
}
```

Response to a Successful Request

```
{
 "_links" : {
  "authReversal" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6461731521426399003473/reversals"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6461731521426399003473"
  },
  "capture" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6461731521426399003473/captures"
  }
 },
 "clientReferenceInformation" : {
  "code" : "1646173152047"
 },
 "id" : "6461731521426399003473",
 "orderInformation" : {
  "amountDetails" : {
   "authorizedAmount" : "100.00",
   "currency" : "usd"
  }
 },
 "paymentAccountInformation" : {
  "card" : {
   "type" : "001"
  }
 },
 "paymentInformation" : {
  "tokenizedCard" : {
   "type" : "001"
  },
  "card" : {
   "type" : "001"
  }
 },
 "paymentInsightsInformation" : {
  "responseInsights" : {
   "categoryCode" : "01"
  }
 },
 "processorInformation" : {
  "systemTraceAuditNumber" : "862481",
  "approvalCode" : "831000",
  "merchantAdvice" : {
   "code" : "01",
   "codeRaw" : "M001"
  },
  "responseDetails" : "ABC",
  "networkTransactionId" : "016153570198200",
  "consumerAuthenticationResponse" : {
   "code" : "2",
      "codeRaw" : "2"
    },
    "transactionId" : "016153570198200",
    "responseCode" : "00",
    "avs" : {
      "code" : "Y",
      "codeRaw" : "Y"
    }
  },
  "reconciliationId" : "64617315214263990O3473",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2024-03-01T22:19:12Z"
}
```

Response to a Declined Request

```
{
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "errorInformation": {
    "reason": "PROCESSOR_ERROR",
    "message": "Invalid account"
  },
  "id": "6583553837826789303954",
  "paymentInsightsInformation": {
    "responseInsights": {
      "categoryCode": "01",
      "category": "ISSUER_WILL_NEVER_APPROVE"
    }
  },
  "pointOfSaleInformation": {
    "amexCapnData": "1009S0600100"
  },
  "processorInformation": {
    "systemTraceAuditNumber": "004544",
    "merchantNumber": "1231231222",
    "networkTransactionId": "431736869536459",
    "transactionId": "431736869536459",
   "responseCode": "111",
    "avs": {
      "code": "Y",
      "codeRaw": "Y"
    }
  },
  "status": "DECLINED"
}
```

Authorizations with Strong Customer Authentication Exemption

This section shows you how to process an authorization with a strong customer authentication (SCA) exemption.

You can use SCA exemptions to streamline the payment process. SCA exemptions are part of the European second Payment Services Directive (PSD2) and allow certain types of low-risk transactions to bypass additional authentication steps while still remaining compliant with PSD2. You can choose which exemption can be applied to a transaction, but the card-issuing bank actually grants an SCA exemption during card authentication.

You can process an authorization with two types of SCA exemptions:

- Exemption on Authorization: Send an authorization without payer authentication and request an SCA exemption on the authorization. If it is not approved, you may be required to request further authentication upon retry.
- Exemption on Authentication: Request an SCA exemption during payer authentication and if successful, send an authorization including the SCA exemption details.

Depending on your processor, use one of these exemption fields:

> **Important**
>
> If you send more than one SCA exemption field with a single authentication, the transaction is denied.

- Authentication Outage: Payer authentication is not available for this transaction due to a system outage.
- B2B Corporate Card: Payment cards specifically for business-to-business transactions are exempt.
- Delegated Authentication: Payer authentication was performed outside of the authorization workflow.
- Follow-On Installment Payment: Installment payments of a fixed amount are exempt after the first transaction.
- Follow-On Recurring Payment: Recurring payments of a fixed amount are exempt after the first transaction.
- Low Risk: The average fraud levels associated with this transaction are considered low.
- Low Value: The transaction value does not warrant SCA.
- Merchant Initiated Transactions: As follow-on transactions, merchant-initiated transactions are exempt.
- Stored Credential Transaction: Credentials are authenticated before storing, so stored credential transactions are exempt.
- Trusted Merchant: Merchants registered as trusted beneficiaries.

Fields Specific to the Strong Customer Authentication Exemptions

Use one of these fields to request an SCA exemption:

consumerAuthenticationInformation. strongAuthentication. authenticationOutageExemptionIndicator	**Exemption type: Authentication Outage** **Value:** 1
consumerAuthenticationInformation. strongAuthentication. secureCorporatePaymentIndicator	**Exemption type: B2B Corporate Card Transaction** **Value:** 1
consumerAuthenticationInformation. strongAuthentication. delegatedAuthenticationExemptionIndicator	**Exemption type: Delegated Authentication** **Value:** 1
consumerAuthenticationInformation. strongAuthentication. riskAnalysisExemptionIndicator	**Exemption type: Low Risk Transaction** **Value:** 1
consumerAuthenticationInformation. strongAuthentication. lowValueExemptionIndicator	**Exemption type: Low Value Transaction** **Value:** 1
consumerAuthenticationInformation. strongAuthentication. trustedMerchantExemptionIndicator	**Exemption type: Trusted Merchant Transaction** **Value:** 1

Country-Specific Requirements

These fields are specific to certain countries and regions.

Argentina

merchantInformation.taxId	Required for Mastercard transactions.
merchantInformation.transactionLocalDateTime	Required when the time zone is not included in your account. Otherwise, this field is optional.

Brazil

paymentInformation.card.sourceAccountType	Required for combo card transactions.
paymentInformation.card.sourceAccountTypeDetails	Required for combo card line-of-credit and prepaid-card transactions.

Chile

merchantInformation.taxId	Required for Mastercard transactions.

Paraguay

merchantInformation.taxId	Required for Mastercard transactions.

Saudi Arabia

processingInformation.authorizationOptions.transactionMode

Taiwan

paymentInformation.card.hashedNumber

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for Processing an Authorization with an SCA Exemption

Use these required fields for processing an authorization that includes an SCA exemption.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

paymentInformation.card.type

Related Information

- *API field reference guide for the REST API*

REST Example: Processing an Authorization with an SCA Exemption for Low-Value Transactions

Request

```
{
 "consumerAutenticationInformation" : {
  "strongAuthentication" : {
   "lowValueExemptionIndicator" : "1"
  }
 },
 "orderInformation" : {
  "billTo" : {
   "country" : "US",
   "lastName" : "Kim",
   "address1" : "201 S. Division St.",
   "postalCode" : "48104-2201",
   "locality" : "Ann Arbor",
   "administrativeArea" : "MI",
   "firstName" : "Kyong-Jin",
   "email" : "test@cybs.com"
  },
  "amountDetails" : {
   "totalAmount" : "100.00",
   "currency" : "eur"
  }
 },
 "paymentInformation" : {
  "card" : {
   "expirationYear" : "2031",
   "number" : "4111111111111111",
   "expirationMonth" : "12"
  }
 }
}
```

Response to a Successful Request

```
{
    "_links": {
        "authReversal": {
            "method": "POST",
            "href": "/pts/v2/payments/6709780221406171803955/reversals"
        },
        "self": {
            "method": "GET",
            "href": "/pts/v2/payments/6709780221406171803955"
        },
        "capture": {
            "method": "POST",
            "href": "/pts/v2/payments/6709780221406171803955/captures"
        }
    },
    "clientReferenceInformation": {
        "code": "1670978022258"
```

```
  },
  "id": "6709780221406171803955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "eur"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "123456"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62859554PBDEMI43",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-12-14T00:33:42Z"
}
```

Zero Amount Authorizations

This section provides the information that you need in order to process a zero amount authorization.

Authorizing a payment for a zero amount shows whether a payment card account is valid and whether the card is lost or stolen. You cannot capture a zero amount authorization.

Processor-Specific Information

Visa Platform Connect	**AVS and CVN are supported.** **Supported for Internet, MOTO, and card-present transactions. Do not try to perform**

a zero amount authorization for a recurring payment, installment payment, or payer authorization transaction.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing a Zero Amount Authorization

Use these required fields for processing a zero amount authorization.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	**Set this value to `0`.**
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
paymentInformation.card.securityCode	

Related Information

- *API field reference guide for the REST API*

Country-Specific Required Fields for Processing a Zero Amount Authorization

Use these country-specific required fields to process a zero amount authorization.

Argentina

merchantInformation.taxId	**Required for Mastercard transactions.**
merchantInformation.transactionLocalDateTime	**Required in Argentina when the time zone is not included in your account. Otherwise, this field is optional.**

Brazil

paymentInformation.card.sourceAccountType	**Required for combo card transactions.**
paymentInformation.card.sourceAccountTypeDetails	**Required for combo card line-of-credit and prepaid-card transactions.**

Saudi Arabia

processingInformation.authorizationOptions.transactionMode

Taiwan

paymentInformation.card.hashedNumber

Related Information

- *API field reference guide for the REST API*

REST Example: Processing a Zero Amount Authorization

Request

```
{
  "orderInformation" : {
    "billTo" : {
      "country" : "US",
      "lastName" : "Kim",
      "address1" : "201 S. Division St.",
      "postalCode" : "48104-2201",
      "locality" : "Ann Arbor",
      "administrativeArea" : "MI",
      "firstName" : "Kyong-Jin",
      "email" : "test@cybs.com"
    },
    "amountDetails" : {
      "totalAmount" : "0.00",
      "currency" : "usd"
```

```
   }
  },
  "paymentInformation" : {
   "card" : {
    "expirationYear" : "2031",
    "number" : "4111111111111111",
    "expirationMonth" : "12"
   }
  }
}
```

Response to a Successful Request

```
{
 "_links" : {
  "authReversal" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6461731521426399003473/reversals"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6461731521426399003473"
  },
  "capture" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6461731521426399003473/captures"
  }
 },
 "clientReferenceInformation" : {
  "code" : "1646173152047"
 },
 "id" : "6461731521426399003473",
 "orderInformation" : {
  "amountDetails" : {
   "authorizedAmount" : "0",
   "currency" : "usd"
  }
 },
 "paymentAccountInformation" : {
  "card" : {
   "type" : "001"
  }
 },
 "paymentInformation" : {
  "tokenizedCard" : {
   "type" : "001"
  },
  "card" : {
   "type" : "001"
  }
 },
 "processorInformation" : {
  "systemTraceAuditNumber" : "862481",
  "approvalCode" : "831000",
  "merchantAdvice" : {
   "code" : "01",
    "codeRaw" : "M001"
   },
   "responseDetails" : "ABC",
   "networkTransactionId" : "016153570198200",
   "consumerAuthenticationResponse" : {
    "code" : "2",
    "codeRaw" : "2"
   },
   "transactionId" : "016153570198200",
   "responseCode" : "00",
   "avs" : {
    "code" : "Y",
    "codeRaw" : "Y"
   }
  },
  "reconciliationId" : "6461731521426399003473",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2022-03-01T22:19:12Z"
 }
```

Pre-Authorizations

This section provides the information you need in order to process a pre-authorization. A pre-authorization enables you to authorize a payment when the final amount is unknown. It is typically used for lodging, auto rental, e-commerce, and restaurant transactions.
For a pre-authorization:

- The authorization amount must be greater than zero.
- The authorization must be submitted for capture within 30 calendar days of its request.
- When you do not capture the authorization, you must reverse it.
  In the U.S., Canada, Latin America, and Asia Pacific, Mastercard charges an additional fee for a pre-authorization that is not captured and not reversed.
  In Europe, Russia, Middle East, and Africa, Mastercard charges fees for all pre-authorizations.
- Chargeback protection is in effect for 30 days after the authorization.

Supported Card Types

All supported card types can process basic authorizations. For a list of all supported card types, see *Payment Processors* on page 20.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for a Pre-Authorization

Use these required fields for processing a pre-authorization.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

Related Information

- *API field reference guide for the REST API*

Country-Specific Required Fields for Processing a Pre-Authorization

Use these country-specific required fields to process a pre-authorization.

Argentina

merchantInformation.taxId	**Required for Mastercard transactions.**
merchantInformation.transactionLocalDateTime	**Required in Argentina when the time zone is not included in your account. Otherwise, this field is optional.**

Brazil

paymentInformation.card.sourceAccountType	**Required for combo card transactions.**

paymentInformation.card.sourceAccountTypeDetails	**Required for combo card line-of-credit and prepaid-card transactions.**

Chile

merchantInformation.taxId	**Required for Mastercard transactions.**

Egypt

paymentInformation.card.cardType	**Required for Meeza transactions. Set to `067`.**
merchantInformation.merchantDescriptor.country	**Required for Meeza transactions. Set to `EG`.**

Paraguay

merchantInformation.taxId	**Required for Mastercard transactions.**

Saudi Arabia

processingInformation.authorizationOptions.transactionMode	**Required only for merchants in Saudi Arabia.**

Taiwan

paymentInformation.card.hashedNumber	**Required only for merchants in Taiwan.**

Related Information

- *API field reference guide for the REST API*

REST Example: Processing a Pre-Authorization

Request

```
{
   "orderInformation": {
      "billTo": {
         "country": "US",
         "lastName": "Kim",
         "address1": "201 S. Division St.",
         "postalCode": "48104-2201",
         "locality": "Ann Arbor",
         "administrativeArea": "MI",
         "firstName": "Kyong-Jin",
         "email": "test@cybs.com"
      },
      "amountDetails": {
         "totalAmount": "100.00",
         "currency": "usd"
      }
```

```
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111111111111111",
      "expirationMonth": "12",
      "type": "001"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links" : {
  "authReversal" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6461731521426399003473/reversals"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6461731521426399003473"
  },
  "capture" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6461731521426399003473/captures"
  }
 },
 "clientReferenceInformation" : {
  "code" : "1646173152047"
 },
 "id" : "6461731521426399003473",
 "orderInformation" : {
  "amountDetails" : {
   "authorizedAmount" : "100.00",
   "currency" : "usd"
  }
 },
 "paymentAccountInformation" : {
  "card" : {
   "type" : "001"
  }
 },
 "paymentInformation" : {
  "tokenizedCard" : {
   "type" : "001"
  },
  "card" : {
   "type" : "001"
  }
 },
 "paymentInsightsInformation" : {
  "responseInsights" : {
   "categoryCode" : "01"
  }
 },
  "processorInformation" : {
    "systemTraceAuditNumber" : "862481",
    "approvalCode" : "831000",
    "merchantAdvice" : {
      "code" : "01",
      "codeRaw" : "M001"
    },
    "responseDetails" : "ABC",
    "networkTransactionId" : "016153570198200",
    "consumerAuthenticationResponse" : {
      "code" : "2",
      "codeRaw" : "2"
    },
    "transactionId" : "016153570198200",
    "responseCode" : "00",
    "avs" : {
      "code" : "Y",
      "codeRaw" : "Y"
    }
  },
  "reconciliationId" : "64617315214263990034 73",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2022-03-01T22:19:12Z"
}
```

Response to a Declined Request

```
{
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "errorInformation": {
    "reason": "PROCESSOR_ERROR",
    "message": "Invalid account"
  },
  "id": "6583553837826789303954",
  "paymentInsightsInformation": {
    "responseInsights": {
      "categoryCode": "01",
      "category": "ISSUER_WILL_NEVER_APPROVE"
    }
  },
  "pointOfSaleInformation": {
    "amexCapnData": "1009S0600100"
  },
  "processorInformation": {
    "systemTraceAuditNumber": "004544",
    "merchantNumber": "1231231222",
    "networkTransactionId": "431736869536459",
    "transactionId": "431736869536459",
   "responseCode": "111",
    "avs": {
      "code": "Y",
      "codeRaw": "Y"
    }
  },
  "status": "DECLINED"
}
```

Incremental Authorizations

This section shows you how to process an incremental authorization.
Incremental authorizations allow merchants to add additional products and services to an existing authorization. This section will show you how to append an original authorization to include additional transactions.
The supported card types for incremental authorizations are Mastercard and Visa.
The incremental authorization service has these limitations:

- Maximum of 100 incremental authorizations per transaction, in addition to the initial authorization.
- Interchange optimization is not supported.
- Split shipments are not supported.

Supported Card Types

These card types support incremental authorizations:

- Mastercard
- Visa

Endpoint

Production: `PATCH https://api.cybersource.com/pts/v2/payments/{id}`
Test: `PATCH https://apitest.cybersource.com/pts/v2/payments/{id}`
The `{id}` is the transaction ID returned in the original authorization response.

Required Fields for Processing an Incremental Authorization

Use these required fields for processing an incremental authorization.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

clientReferenceInformation.code

orderInformation.amountDetails.additionalAmount

orderInformation.amountDetails.currency

processingInformation.authorizationOptions.initiator.storedCredentialUsed	Set this field to `true`.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

Related Information

- *API field reference guide for the REST API*

Country-Specific Required Fields for Processing an Incremental Authorization

Use these country-specific required fields to process an incremental authorization.

Argentina

merchantInformation.transactionLocalDateTime	**Required in Argentina when the time zone is not included in your account. Otherwise, this field is optional.**

Related Information

- *API field reference guide for the REST API*

Optional Field for Processing an Incremental Authorization

You can use this optional field to include your transaction ID when processing an incremental authorization.

clientReferenceInformation.transactionId

Related Information

- *API field reference guide for the REST API*

REST Example: Processing an Incremental Authorization

Request

```
{
  "clientReferenceInformation": {
    "code": "33557799"
  },
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "storedCredentialUsed": "true"
   }
  }
 },
 "orderInformation" : {
  "amountDetails" : {
   "additionalAmount": "105.00",
   "currency" : "USD"
  }
 }
 "merchantInformation": {
  "transactionLocalDateTime": "20191002080000"
 }
}
```

Response to a Successful Request

```
{
 "_links" : {
  "authReversal" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6479624584536070903093/reversals"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6479624584536070903093"
  },
  "capture" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6479624584536070903093/captures"
  }
 },
 "clientReferenceInformation" : {
  "code" : "33557799"
 },
 "id" : "6479624584536070903093",
 "orderInformation" : {
  "amountDetails" : {
   "authorizedAmount" : "105.00",
   "currency" : "USD"
  }
 },
 "paymentAccountInformation" : {
  "card" : {
   "type" : "001"
  }
 },
 "paymentInformation" : {
  "tokenizedCard" : {
   "type" : "001"
  },
  "card" : {
   "type" : "001"
```

```
    }
  },
  "processorInformation" : {
    "systemTraceAuditNumber" : "819203",
    "approvalCode" : "831000",
    "cardVerification" : {
      "resultCodeRaw" : "M",
      "resultCode" : "M"
    },
    "merchantAdvice" : {
      "code" : "01",
      "codeRaw" : "M001"
    },
    "responseDetails" : "ABC",
    "networkTransactionId" : "016153570198200",
    "retrievalReferenceNumber" : "208115819203",
    "consumerAuthenticationResponse" : {
      "code" : "2",
      "codeRaw" : "2"
    },
    "transactionId" : "016153570198200",
    "responseCode" : "00",
    "avs" : {
      "code" : "Y",
      "codeRaw" : "Y"
    }
  },
  "reconciliationId" : "6479624584536070903093",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2022-03-22T15:20:58Z"
}
```

Final Authorization Indicator

The purpose of this feature is to ensure that unused funds are reversed, so that customer’s funds are available again when an order is not fulfilled.

For an authorization with an amount greater than zero, indicate whether the authorization is a final authorization, a pre-authorization, or an undefined authorization.

You can set a default authorization type in your account. To set the default authorization type in your account, contact customer support.

Chargeback protection is in effect for seven days after the authorization.

Supported Services

- Authorization
- Incremental authorization

Supported Card Types

- Co-badged Mastercard and mada. You must identify the card type as Mastercard. Supported only on Visa Platform Connect.
- Maestro (International)
- Maestro (UK Domestic)
- Mastercard

Requirements for Final Authorizations

For a final authorization:

- The authorization amount must be greater than zero.
- The authorization amount must be the final amount that the customer agrees to pay.
- The authorization should not be cancelled after it is approved except when a system failure occurs.
- The authorization must be submitted for capture within seven calendar days of its request.
- The capture amount and currency must be the same as the authorization amount and currency.

Pre-Authorizations

A pre-authorization enables you to authorize a payment when the final amount is unknown. It is typically used for lodging, auto rental, e-commerce, and restaurant transactions.

For a pre-authorization:

- The authorization amount must be greater than zero.
- The authorization must be submitted for capture within 30 calendar days of its request.
- When you do not capture the authorization, you must reverse it.

  In the U.S., Canada, Latin America, and Asia Pacific, Mastercard charges an additional fee for a pre-authorization that is not captured and not reversed.

  In Europe, Russia, Middle East, and Africa, Mastercard charges fees for all pre-authorizations.
- Chargeback protection is in effect for 30 days after the authorization.

Unmarked Authorizations

An authorization is unmarked when the default authorization type is not set in your account and you do not include the **authIndicator** field in the authorization request. To set the default authorization type in your account, contact customer support.

Unmarked authorizations are supported only in the US, Canada, Latin America, and Asia Pacific. They are not supported in Europe, Russia, Middle East, and Africa.

Cybersource does not set a mark or indicator for the type of authorization in the request that is sent to the processor.

> **Important**
>
> Your acquirer processes an unmarked authorization as a final authorization, a preauthorization, or an undefined authorization. Contact your acquirer to learn how they process unmarked authorizations.

Requirements for Unmarked Authorizations

For an unmarked authorization:

- The authorization amount must be greater than zero.
- The authorization amount can be different from the final transaction amount.

Undefined Authorizations

An authorization is undefined when you set the default authorization type in your account to undefined and do not include the **authIndicator** field in the authorization request. To set the default authorization type in your account, contact customer support.

Undefined authorizations are supported only in the U.S., Canada, Latin America, and Asia Pacific. They are not supported in Europe, Russia, Middle East, and Africa.

Chargeback protection is in effect for seven days after the authorization.

Requirements for Undefined Authorizations

For an undefined authorization:

- The authorization amount must be greater than zero.
- The authorization amount can be different from the final transaction amount.
- The authorization should not be cancelled after it is approved except when a system failure occurs.
- The authorization must be submitted for capture within seven calendar days of its request.
- When you do not capture the authorization, you must reverse it; otherwise, Mastercard charges an additional fee for the transaction.

Required Fields for Final Authorizations

Use these required fields for final authorizations and preauthorizations.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.type	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.authIndicator	**Set the value to `0` for preauthorizations, or to `1` for final authorizations. Do not include this field for unmarked or undefined authorizations.**

REST Example: Final Authorizations

Request

```
{
 "orderInformation" : {
  "billTo" : {
   "firstName" : "RTS",
   "lastName" : "VDP",
   "address1" : "201 S. Division St.",
   "postalCode" : "48104-2201",
   "locality" : "Ann Arbor",
   "administrativeArea" : "MI",
   "country" : "US",
   "email" : "test@cybs.com"
  },
  "amountDetails" : {
   "totalAmount" : "100.00",
   "currency" : "usd"
  }
 },
 "paymentInformation" : {
  "card" : {
   "expirationYear" : "2031",
   "number" : "4111111111111111",
   "expirationMonth" : "12",
   "type" : "001"
  }
 },
 "processingInformation" : {
  "authorizationOptions" : {
    "authIndicator" : "1"
  }
 }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6910040807416719003955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6910040807416719003955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6910040807416719003955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1691004080800"
  },
  "id": "6910040807416719003955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "usd"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "67628631TKRG2OVE",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-08-02T19:21:20Z"
}
```

Authorization Reversal

This section provides the information about how to process an authorization reversal. Reversing an authorization releases the hold on the customer's payment card funds that the issuing bank placed when processing the authorization.

For a debit card or prepaid card in which only a partial amount was approved, the amount of the reversal must be the amount that was authorized, not the amount that was requested.

Supported Card Types

All supported card types can process reversals.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/reversals`

Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/reversals`

The `{id}` is the transaction ID returned in the authorization response.

Required Fields for Processing an Authorization Reversal

clientReferenceInformation.code	
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
reversalInformation.amountDetails.currency	
reversalInformation.amountDetails.totalAmount	The amount of the reversal must be the same as the authorization amount that was included in the authorization response message. Do not use the amount that was requested in the authorization request message.

REST Example: Processing an Authorization Reversal

Request

```
{
  "clientReferenceInformation": {
   "code": "test123"
  }
  "reversalInformation" : {
    "amountDetails" : {
      "totalAmount" : "100.00",
      "currency" : "USD"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links" : {
   "self" : {
     "method" : "GET",
     "href" : "/pts/v2/reversals/6869460219566537303955"
   }
  },
  "clientReferenceInformation" : {
    "code" : "RTS-Auth-Reversal"
  },
  "id" : "6869460219566537303955",
  "orderInformation" : {
    "amountDetails" : {
      "currency" : "USD"
    }
  },
  "processorInformation" : {
    "responseCode" : "200"
  },
  "reconciliationId" : "82kBK3qDNtls",
  "reversalAmountDetails" : {
    "reversedAmount" : "100.00",
    "currency" : "USD"
  },
  "status" : "REVERSED",
  "submitTimeUtc" : "2023-06-16T20:07:02Z"
}
```

Time-Out Authorization Reversals

When you do not receive a response message after sending an authorization request, this feature enables you to reverse the authorization that you requested.

> **Important**
>
> Wait 60 seconds before requesting a time-out authorization reversal.

Include the **clientReferenceInformation.transactionId** field in the original request for an authorization. The value of the merchant transaction ID must be unique for 180 days. When the original transaction fails, the response message for the reversal request includes these fields:

- **reversalAmountDetails.originalTransactionAmount**
- **processorInformation.responseCode**

Requirements

Unless your processor supports authorization reversal after void (ARAV), time-out authorization reversals are supported only for authorizations that have not been captured and settled.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/reversals`
Test: `POST https://apitest.cybersource.com/pts/v2/reversals`

Required Fields for Processing a Time-Out Authorization Reversal

Use these required fields for processing a time-out authorization reversal.

clientReferenceInformation.transactionId	Identifier that links the reversal request to the original request.
reversalInformation.amountDetails.currency	
reversalInformation.amountDetails.totalAmount	The amount of the reversal must be the same as the authorization amount that was included in the authorization response message. Do not use the amount that was requested in the authorization request message.

Related Information

- *REST API Field Reference*

REST Example: Processing a Time-Out Authorization Reversal

Request

```
{
  "clientReferenceInformation": {
    "transactionId": "987654321"
 },
   "reversalInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    },
    "reason": "testing"
  }
}
```

Response to Successful Request

```
{
 "_links" : {
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/reversals/6869460219566537303955"
  }
 },
 "clientReferenceInformation" : {
  "code" : "RTS-Auth-Reversal"
 },
 "id" : "6869460219566537303955",
 "orderInformation" : {
  "amountDetails" : {
   "currency" : "USD"
  }
 },
 "processorInformation" : {
  "responseCode" : "200"
 },
 "reconciliationId" : "82kBK3qDNtls",
 "reversalAmountDetails" : {
  "reversedAmount" : "100.00",
  "currency" : "USD"
 },
 "status" : "REVERSED",
 "submitTimeUtc" : "2023-06-16T20:07:02Z"
}=
```

Sale

This section provides the information you need in order to process a sale transaction. A sale combines an authorization and a capture into a single transaction.

Supported Card Types

All supported card types can process sales. .

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing a Sale

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
paymentInformation.card.securityCode	
paymentInformation.card.type	
processingInformation.capture	Set the value to `true`.

Related Information

- *API field reference guide for the REST API*

REST Example: Processing a Sale

Request

```
{
  "processingInformation": {
    "capture": true
  },
  "orderInformation" : {
    "billTo" : {
    "country" : "US",
    "lastName" : "VDP",
    "address1" : "201 S. Division St.",
    "postalCode" : "48104-2201",
    "locality" : "Ann Arbor",
    "administrativeArea" : "MI",
    "firstName" : "RTS",
    "email" : "test@cybs.com"
  },
    "amountDetails" : {
      "totalAmount" : "100.00",
      "currency" : "usd"
     }
   },
  "paymentInformation" : {
    "card" : {
      "expirationYear" : "2031",
      "number" : "4111111111111111",
   "expirationMonth" : "12",
   "type" : "001
  }
 }
}
```

Response to a Successful Request

Most processors do not return all of the fields that are shown in this example.

```
{
 "_links" : {
  "void" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6485004068966546103093/voids"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6485004068966546103093"
  }
 },
 "clientReferenceInformation" : {
  "code" : "RTS-Auth"
 },
 "id" : "6485004068966546103093",
 "orderInformation" : {
  "amountDetails" : {
   "totalAmount" : "100.00",
   "authorizedAmount" : "100.00",
   "currency" : "usd"
  }
 },
 "paymentAccountInformation" : {
  "card" : {
   "type" : "001"
  }
 },
 "paymentInformation" : {
  "tokenizedCard" : {
   "type" : "001"
  },
  "card" : {
   "type" : "001"
  }
 },
 "processorInformation" : {
  "systemTraceAuditNumber" : "841109",
  "approvalCode" : "831000",
  "merchantAdvice" : {
   "code" : "01",
   "codeRaw" : "M001"
  },
  "responseDetails" : "ABC",
  "networkTransactionId" : "016153570198200",
  "retrievalReferenceNumber" : "208720841109",
  "consumerAuthenticationResponse" : {
      "code" : "2",
      "codeRaw" : "2"
    },
    "transactionId" : "016153570198200",
    "responseCode" : "00",
    "avs" : {
      "code" : "Y",
      "codeRaw" : "Y"
    }
  },
  "reconciliationId" : "6485004068966546103093",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2022-03-28T20:46:47Z"
}
```

Sales with Payment Network Tokens

This section shows you how to successfully process a sale with payment network tokens.

Important

Due to mandates from the Reserve Bank of India, Indian merchants cannot store personal account numbers (PAN). Use network tokens instead. For more information on network tokens, see *Network Tokenization* in the Token Management Service Developer Guide.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for Sales with Payment Network Tokens

Use these required fields for processing a sale with payment network tokens.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
paymentinformation.tokenizedCard.cryptogram	
paymentinformation.tokenizedCard.expirationMonth	
paymentinformation.tokenizedCard.expirationYear	
paymentInformation.tokenizedCard.transactionType	
processingInformation.capture	**Set the value to `true`.**

Related Information

- *API field reference guide for the REST API*

Optional Fields for Sales with Payment Network Tokens

You can use these optional fields to include additional information when processing a sale with a payment network token.

clientReferenceInformation.code	
consumerAuthenticationInformation.cavv	**For 3-D Secure in-app transactions for Visa and JCB, set this field to the 3-D Secure cryptogram. Otherwise, set to the network token cryptogram.**
consumerAuthenticationInformation.ucafAuthenticationData	**For Mastercard requests using 3-D Secure, set this field to the Identity Check cryptogram.**
consumerAuthenticationInformation.ucafCollectionIndicator	**For Mastercard requests using 3-D Secure, set the value to `2`.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	**Required only for transactions in the US and Canada.**

orderInformation.billTo.administrativeArea	**Required only for transactions in the US and Canada.**
processingInformation.commerceIndicator	
paymentInformation.tokenizedCard.cardType	**It is strongly recommended that you send the card type even if it is optional for your processor. Omitting the card type can cause the transaction to be processed with the wrong card type.**
paymentInformation.tokenizedCard.cryptogram	
paymentInformation.tokenizedCard.expirationMonth	**Set to the token expiration month that you received from the token service provider.**
paymentInformation.tokenizedCard.expirationYear	**Set to the token expiration year that you received from the token service provider.**
paymentInformation.tokenizedCard.number	**Set to the token value that you received from the token service provider.**
paymentInformation.tokenizedCard.requestorId	**Required on Visa Platform Connect**
paymentInformation.tokenizedCard.transactionType	

Related Information

- *API field reference guide for the REST API*

REST Example: Sales with Payment Network Tokens

Request

```
{
  "orderInformation" : {
    "billTo": {
      "country": "US",
      "lastName": "Kim",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "firstName": "Smith",
      "email": "test@cybs.com"
    },
    "amountDetails" : {
      "totalAmount" : "100",
      "currency" : "USD"
    }
  },
    "paymentInformation" : {
    "tokenizedCard" : {
      "expirationYear" : "2031",
      "number" : "4111111111111111",
    "expirationMonth" : "12",
    "transactionType" : "1",
    "cryptogram" : "qE5juRwDzAUFBAkEHuWW9PiBkWv="
   }
  }
 }
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6838294805206235603954/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6838294805206235603954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6838294805206235603954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1683829480593"
  },
  "id": "6838294805206235603954",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "1"
    }
  },
  "reconciliationId": "60332034UHI9PRJ0",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-05-11T18:24:40Z"
}
```

Captures

This section provides the information you need in order to capture an authorized transaction.

Supported Card Types

All supported card types can process captures. .

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/captures`
Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures`
The `{id}` is the transaction ID returned in the authorization response.

Required Fields for Capturing an Authorization

Use these required fields for capturing an authorization.

clientReferenceInformation.code	This field value maps from the original authorization, sale, or credit transaction.
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	

REST Example: Capturing an Authorization

Request

```
{
  "clientReferenceInformation": {
    "code": "ABC123"
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "EUR"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/captures/6662994431376681303954/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/captures/6662994431376681303954"
    }
  },
  "clientReferenceInformation": {
    "code": "1666299443215"
  },
  "id": "6662994431376681303954",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "EUR"
    }
  },
  "reconciliationId": "66535942B9CGT52U",
  "status": "PENDING",
  "submitTimeUtc": "2022-10-20T20:57:23Z"
}
```

Marketplace Captures with Foreign Retailers

Visa Platform Connect requires marketplaces to identify foreign retail transactions when the marketplace and issuer are in the European Economic Area (EEA), the U.K., and Gibraltar and the retailer is in a different country. For marketplace transactions, the marketplace is the merchant and the retailer is the sub-merchant. Marketplace foreign retail transactions are identified in the Business Center on the transactions details page.

> **Important**
>
> The capture request data overrides the authorization request data.

Fields Specific to this Use Case

These fields are required for this use case:

aggregatorInformation.subMerchant.country	Set this value to the retailer country.
merchantInformation.merchantDescriptor.country	Set this value to the marketplace country.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/captures`

Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures`
The `{id}` is the transaction ID returned in the authorization response.

Required Fields for Capturing an Authorization with a Foreign Retailer

Use these required fields for capturing an authorization.

aggregatorInformation.subMerchant.country	**Set this field to the retailer country.**
clientReferenceInformation.code	**This field value maps from the original authorization, sale, or credit transaction.**
clientReferenceInformation.partner.thirdPartyCertificationNumber	**Cybersource provides the value for this field.**
merchantInformation.merchantDescriptor.country	**Set this field to the marketplace country.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	

REST Example: Capturing a Marketplace Authorization with a Foreign Retailer

Request

```
{
   "aggregatorInformation" : {
      "subMerchant" : {
         "country" : "AU"
      }
   },{
   "clientReferenceInformation": {
      "code": "ABC123",
      "partner": {
         "thirdPartyCertificationNumber": "123456789012"
      }
   {
   "merchantInformation" : {
      "merchantDescriptor" : {
         "country" : "GB"
      }
   },   },
   "orderInformation": {
      "amountDetails": {
         "totalAmount": "100.00",
         "currency": "GBP"
   }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/captures/6662994431376681303954/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/captures/6662994431376681303954"
    }
  },
  "clientReferenceInformation": {
    "code": "1666299443215"
  },
  "id": "6662994431376681303954",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "GBP"
    }
  },
  "reconciliationId": "66535942B9CGT52U",
  "status": "PENDING",
  "submitTimeUtc": "2024-10-20T20:57:23Z"
}
```

Multiple Partial Captures

This section shows you how to process multiple partial captures for an authorization.

This feature enables you to request multiple partial captures for one authorization. A multiple partial capture allows you to incrementally settle authorizations over time. Ensure that the total amount of all the captures does not exceed the authorized amount.

Fields Specific to This Use Case

These API request fields and values are specific to this use case:

processingInformation.captureOptions.captureSequenceNumber

processingInformation.captureOptions.totalCaptureCount

Prerequisite

Contact customer support to have your account enabled for this feature.

Limitations

Your account can be enabled for multiple partial captures or split shipments; it cannot be enabled for both features.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments/{id}/captures
Test: POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures
The {id} is the transaction ID returned in the authorization response.

Required Fields for Processing Multiple Partial Captures

clientReferenceInformation.code	Set to **clientReferenceInformation.code** value used in corresponding authorization request.
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
processingInformation.captureOptions.captureSequenceNumber	For the final capture request, set this field and **processingInformation.captureOptions.totalCaptureCo** to the same value.
processingInformation.captureOptions.totalCaptureCount	When you do not know the total number of captures that you are going to request, set this field to at least one more than the **processingInformation.captureOptions.captureSequenceNumber** field until you reach the final capture. For the final capture request, set this field and **processingInformation.captureOptions.captureSequenceNumber** to the same value.

Related Information

- *API field reference guide for the REST API*

REST Example: Processing Multiple Partial Captures

Request

```
{
 {
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "processingInformation": {
  "captureOptions": {
   "captureSequenceNumber": "2",
   "totalCaptureCount": "3"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
   "href": "/pts/v2/captures/6742496815656503003954/voids"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/captures/6742496815656503003954"
  }
 },
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "id": "6742496815656503003954",
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 },
 "reconciliationId": "67332020GD2G1OO1",
 "status": "PENDING",
 "submitTimeUtc": "2023-01-20T21:21:21Z"
}
```

Forced Captures

This feature allows merchants to process authorizations obtained through an organization other than Cybersource. For example, a merchant might call their processor to request a manual authorization, at which point they can request a forced capture of the authorization.

A manual authorization cannot be captured for more than the original authorization amount, and the authorization expires after seven days.

Supported Acquirers

- Banco Safra
- Bank Sinarmas (Omise Ltd.)

- BC Card Co., Ltd.
- Citibank Malaysia
- CTBC Bank Ltd.
- Sumitomo Mitsui Card Co.
- Vietnam Technological and Commercial Joint-Stock Bank

Supported Services

- Authorization

Required Fields for Forced Captures

Use these required fields for processing forced captures.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.type	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.authType	Set the value to `verbal`.
processingInformation.authorizationOptions.verbalAuthCode	Set this field to the manually obtained authorization code.

Related Information

- *API field reference guide for the REST API*

REST Example: Forced Captures

Request

```
{
  "orderInformation": {
    "billTo" : {
      "firstName" : "RTS",
      "lastName" : "VDP",
      "address1" : "201 S. Division St.",
      "postalCode" : "48104-2201",
      "locality" : "Ann Arbor",
      "administrativeArea" : "MI",
      "country" : "US",
      "email" : "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "usd"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111111111111111",
      "expirationMonth": "12",
      "type": "001"
    }
  },
  "processingInformation": {
    "authorizationOptions": {
      "authType": "verbal",
      "verbalAuthCode": "ABC123"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6915126171696653403954/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6915126171696653403954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6915126171696653403954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "id": "6915126171696653403954",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "ABC123"
  },
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-08-08T16:36:57Z"
}
```

Refunds

This section provides the information you need in order to process a refund, which is linked to a capture or sale. You must request a refund within 180 days of the authorization.

When your account is enabled for credit authorizations, also known as purchase return authorizations, Cybersource authenticates the card and customer during a refund or credit request. Every credit request is automatically authorized.

Credit authorization results are returned in these response fields:

- **processorInformation.approvalCode**
- **processorInformation.networkTransactionId**
- **processorInformation.responseCode**

When you request a void for the credit and the credit is voided. If your account is enabled for credit authorizations, the credit authorization is also reversed.

Supported Card Types

All supported card types can process refunds. For a list of all supported card types, see *Payment Processors* on page 20.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/refunds`

Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/refunds`

The `{id}` is the transaction ID returned in the capture or sale response.

Required Fields for Processing a Refund

Use these required fields for processing a refund.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

Related Information

- *API field reference guide for the REST API*

REST Interactive Example: Processing a Refund

Refund a Payment

Live Console URL: *https://developer.cybersource.com/api-reference-assets/index.html#payments_refund_refund-a-payment*

REST Example: Processing a Refund

Request

```
{
   "orderInformation": {
      "amountDetails": {
         "totalAmount": "100.00",
         "currency": "EUR"
      }
   }
}
```

Response to a Successful Request

```
{
   "_links": {
      "void": {
         "method": "POST",
         "href": "/pts/v2/credits/6699964581696622603955/voids"
      },
      "self": {
         "method": "GET",
         "href": "/pts/v2/credits/6699964581696622603955"
      }
   },
   "clientReferenceInformation": {
      "code": "1669996458298"
   },
   "creditAmountDetails": {
      "currency": "eur",
      "creditAmount": "100.00"
   },
   "id": "6699964581696622603955",
   "orderInformation": {
    "amountDetails": {
      "currency": "EUR"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "016153570198200",
    "responseCode": "100"
  },
  "reconciliationId": "61873329OAILG3Q6",
  "status": "PENDING",
  "submitTimeUtc": "2022-12-02T15:54:18Z"
}
```

Credits

This section shows you how to process a credit, which is not linked to a capture or sale. There is no time limit for requesting a credit.

When your account is enabled for credit authorizations, also known as purchase return authorizations, Cybersource authenticates the card and customer during a refund or credit request. Every credit request is automatically authorized.

Credit authorization results are returned in these response fields:

- **processorInformation.approvalCode**
- **processorInformation.networkTransactionId**
- **processorInformation.responseCode**

When you request a void for the credit and the credit is voided. If your account is enabled for credit authorizations, the credit authorization is also reversed.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/credits/

Test: POST https://apitest.cybersource.com/pts/v2/credits/

Required Fields for Processing a Credit

Use these required fields for processing a credit.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

REST Interactive Example: Processing a Credit

```
Credit
```

Live Console URL: *https://developer.cybersource.com/api-reference-assets/index.html#payments_credit_process-a-credit*

REST Example: Processing a Credit

Request

```
{
  "orderInformation" : {
    "billTo" : {
      "country" : "US",
      "lastName" : "Kim",
      "address1" : "201 S. Division St.",
      "postalCode" : "48104-2201",
      "locality" : "Ann Arbor",
      "administrativeArea" : "MI",
      "firstName" : "Kyong-Jin",
      "email" : "test@cybs.com"
  },
  "amountDetails" : {
   "totalAmount" : "100.00",
   "currency" : "eur"
  }
 },
 "paymentInformation" : {
  "card" : {
   "expirationYear" : "2031",
   "number" : "4111111111111111",
   "expirationMonth" : "12"
  }
 }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/credits/6663069906146706403954/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/credits/6663069906146706403954"
    }
  },
  "clientReferenceInformation": {
    "code": "1666306990717"
  },
  "creditAmountDetails": {
    "currency": "eur",
    "creditAmount": "100.00"
  },
  "id": "6663069906146706403954",
  "orderInformation": {
    "amountDetails": {
      "currency": "eur"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "016153570198200",
    "responseCode": "100"
  },
  "reconciliationId": "66490108K9CLFJPN",
  "status": "PENDING",
  "submitTimeUtc": "2022-10-20T23:03:10Z"
}
```

Voids for a Capture or Credit

This section describes how to void a capture or credit that was submitted but not yet processed by the processor.

Endpoint

Void a Capture
Production: `POST https://api.cybersource.com/pts/v2/captures/{id}/voids`
Test: `POST https://apitest.cybersource.com/pts/v2/captures/{id}/voids`
Void a Credit
Production: `POST https://api.cybersource.com/pts/v2/credits/{id}/voids`
Test: `POST https://apitest.cybersource.com/pts/v2/credits/{id}/voids`
The `{id}` is the transaction ID returned during the capture or credit response.

Required Fields for Voiding a Capture or Credit

clientReferenceInformation.code	**Including this field is recommended, but not required.**

REST Example: Voiding a Capture or Credit

Request

```
{
  "clientReferenceInformation": {
    "code": "test123"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/voids/6541933390746728203005"
    }
  },
  "clientReferenceInformation": {
    "code": "1654193339056"
  },
  "id": "6541933390746728203005",
  "orderInformation": {
    "amountDetails": {
    "currency": "USD"
    }
  },
  "status": "VOIDED",
  "submitTimeUtc": "2022-06-02T18:08:59Z",
  "voidAmountDetails": {
    "currency": "usd",
    "voidAmount": "100.00"
  }
}
```

Time-Out Voids for a Capture, Sale, Refund, or Credit

When you do not receive a response message after requesting a capture, sale, refund, or credit, this feature enables you to void the transaction that you requested.

Include the **clientReferenceInformation.transactionId** field in the original request for a capture, sale, refund, or credit. The value of the merchant transaction ID must be unique for 180 days.

When the original transaction fails, the response message for the reversal request includes these fields:

- **voidAmountDetails.originalTransactionAmount**
- **processorInformation.responseCode**

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/voids/`

Test: `POST https://apitest.cybersource.com/pts/v2/voids/`

Required Fields for a Time-Out Void for a Capture, Sale, Refund, or Credit

clientReferenceInformation.transactionId

Related Information

- *API field reference guide for the REST API*

REST Example: Time-Out Void for a Capture, Sale, Refund, or Credit

Request

```
{
 "clientReferenceInformation": {
    "transactionId": "987654321"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "self": {
      "method": "GET",
        "href": "/pts/v2/voids/6541933390746728203005"
    }
  },
  "clientReferenceInformation": {
    "code": "1654193339056"
  },
  "id": "6541933390746728203005",
  "orderInformation": {
    "amountDetails": {
      "currency": "USD"
    }
  },
  "status": "VOIDED",
  "submitTimeUtc": "2022-06-02T18:08:59Z",
  "voidAmountDetails": {
    "currency": "usd",
    "voidAmount": "100.00"
  }
}
```

Request

```
{
  "clientReferenceInformation": {
    "transactionId": "1234231332213112"
  }
}
```

Card Present Connect | Retail Processing

This section shows you how to process these card-present transactions:

- Authorizations with contact EMV and an online PIN
- Authorizations with contact EMV and an offline PIN
- Authorizations with contactless EMV and an online PIN
- Authorizations with magnetic stripe swipe
- Authorizations with hand-keyed data
- Authorizations for a cash advance with a credit card
- Capturing an authorization

Additional Resources for Card Present Connect | Retail

For more information on Card Present Connect | Retail, see these guides:

- *Card Present Connect | Retail Guide*
- *API field reference guide for the REST API*
- Github repositories: *https://github.com/Cybersource*

Retail EMV and Card Data

You can request these payment services for retail with EMV and card data:

- Authorization: standard and incremental
- Capture
- Stand-alone credit

This table shows which EMV tags are:

- M: mandatory
- P: prohibited
- O: optional
- C: conditional (Send the tag when it is present in card and terminal.)

EMV Data Elements and Tags

Data Element	EMV Tag	Mastercard	Visa
Transaction Date	9A	M	M
Transaction Type	9C	M	M
Transaction Currency Code	5F2A	M	M
Terminal Country Code	9F1A	M	M
Amount Authorized	9F02	M	M
Amount Other	9F03	M	M
Application PAN Sequence Number	5F34	C	O
Application Transaction Counter (ATC)	9F36	M	M
Application Interchange Profile (AIP)	82	M	M
Dedicated File (DF) Name	84	M	M
Terminal Verification Results (TVR)	95	M	M
Issuer Application Data	9F10	M	M
Application Cryptogram	9F26	M	M
Cryptogram Information Data (CID)	9F27	M	O
Terminal Capabilities	9F33	M	M
Cardholder Verification Method (CVM) Results	9F34	M	O
Unpredictable Number (UN)	9F37	M	M
Form Factor Indicator	9F6E	O (Authorization) P (Refund)	C

Authorization with Contact EMV and Online PIN

For an EMV chip contact authorization, the customer inserts the card directly into a point-of-sale (POS) terminal. For an online PIN authorization, the customer enters a PIN to verify their identity, and the issuer verifies the PIN.
Online PIN transactions are supported by these card types:

- Visa
- Mastercard
- American Express
- Discover

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing an Authorization with Contact EMV and Online PIN

clientReferenceInformation.code	
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
merchantInformation.transactionLocalDateTime	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	
pointOfSaleInformation.emv.cardSequenceNumber	
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.encryptedKeySerialNumber	
pointOfSaleInformation.encryptedPin	
pointOfSaleInformation.entryMode	Set the value to `contact` for an EMV payment.
pointOfSaleInformation.pinBlockEncodingFormat	
pointOfSaleInformation.terminalCapability	Set the value to `4`.
pointOfSaleInformation.terminalPinCapability	

pointOfSaleInformation.trackData	
processingInformation.commerceIndicator	**Set the value to `retail`.**

Country-Specific Required Fields for Processing an Authorization with Contact EMV or Contactless PIN

Argentina

merchantInformation.transactionLocalDateTime	Required when the time zone is not set in your account.
invoiceDetails.salesSlipNumber	

India

pointOfSaleInformation.terminalCompliance	

Japan

invoiceDetails.salesSlipNumber	

REST Example: Processing an Authorization with Contact EMV and Online PIN

Request

```
{
 "clientReferenceInformation": {
  "code": "test123",
  "transactionId": "uniqueValue1",
  "partner": {
   "thirdPartyCertificationNumber": "testTPCN"
  }
 },
 "processingInformation": {
  "commerceIndicator": "retail",
  }
 },
 "paymentInformation": {
   "card": {
        "type": "001"
    }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "9900.00",
   "currency": "USD"
  }
 },
 "pointOfSaleInformation": {
  "entryMode": "contact",
    "terminalCapability": 4,
     "terminalPinCapability": 6,
    "emv": {
      "tags":
  "5F3401019F3303E0F8C895058080048009F370465B81A3A9F100706011203A0A0009F2608E9D097D1901E8AB99F36020002
      "cardSequenceNumber": "01"
    },
    "trackData": ";4761xxxxxxxxxxxx=251220111478549?",
    "pinBlockEncodingFormat":0,
    "encryptedPin": "F509429A3C3FD201",
    "encryptedKeySerialNumber": "FFFF1B1D140000200001"
  },
   "merchantInformation": {
  "transactionLocalDateTime": "20230724085022"
 }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6938891699856080004953/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6938891699856080004953"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6938891699856080004953/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "test123",
    "transactionId": "uniqueValue1"
  },
  "id": "6938891699856080004953",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "9900.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "accountFeatures": {
      "category": "A",
      "group": "0"
    },
    "tokenizedCard": {
```

```
    "type": "001"
   },
   "card": {
    "type": "001"
   }
  },
  "pointOfSaleInformation": {
   "emv": {
    "tags":
 "9F3602001591 0AB58D60185BEF0247303072179F18043030303186 0E04DA9F580903B1BAEDFD1438BA48"
   }
  },
  "processorInformation": {
   "systemTraceAuditNumber": "188535",
   "approvalCode": "831000",
   "networkTransactionId": "016153570198200",
   "retrievalReferenceNumber": "324704188535",
   "transactionId": "016153570198200",
   "responseCode": "00",
   "avs": {
    "code": "2"
   }
  },
  "reconciliationId": "6938891699856080004953",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-09-05T04:46:10Z"
}
```

Authorization with Contact EMV and Offline PIN

During a contact EMV authorization, the customer inserts the card into the terminal, which causes the EMV chip to be in contact with the terminal. When processing an offline PIN transaction, the EMV chip verifies the customer PIN.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for Processing an Authorization with Contact EMV and Offline PIN

clientReferenceInformation.code	
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	

merchantInformation.transactionLocalDateTime	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	
pointOfSaleInformation.emv.cardSequenceNumber	
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	**Set the value to `contact`.**
pointOfSaleInformation.terminalCapability	**Set the value to `4`. Set the value to `0` if the terminal does not support PINs.**
pointOfSaleInformation.terminalPinCapability	
pointOfSaleInformation.trackData	
processingInformation.commerceIndicator	**Set the value to `retail`.**

Country-Specific Required Fields for Processing an Authorization with Contact EMV or Contactless PIN

Argentina

merchantInformation.transactionLocalDateTime	Required when the time zone is not set in your account.
invoiceDetails.salesSlipNumber	

India

pointOfSaleInformation.terminalCompliance

Japan

invoiceDetails.salesSlipNumber

REST Example: Processing an Authorization with Contact EMV and Offline PIN

Request

```
{
  "clientReferenceInformation": {
    "code": "test123",
    "transactionId": "uniqueValue2",
    "partner": {
      "thirdPartyCertificationNumber": "testTPCN"
    }
  },
  "processingInformation": {
   "commerceIndicator": "retail",
   "authorizationOptions": {
     "partialAuthIndicator": "true"
   }
  },
  "paymentInformation": {
    "card": {
          "type": "001"
     }
  },
  "orderInformation": {
   "amountDetails": {
     "totalAmount": "9900.00",
     "currency": "USD"
   }
  },
  "pointOfSaleInformation": {
   "entryMode": "contact",
   "terminalCapability": 4,
    "terminalPinCapability": 6,
   "emv": {
     "tags":
 "5F3401019F3303E0F8C8950580800480009F370465B81A3A9F100706011203A0A0009F2608E9D097D1901E8AB99F36020002
     "cardSequenceNumber": "01"
   },
   "trackData": ";4761xxxxxxxxxxxx=251220111478549?"
  },
   "merchantInformation": {
  "transactionLocalDateTime": "20230724085022"
 }
}
```

Response to a Successful Request

```
{
  "_links": {
   "authReversal": {
     "method": "POST",
     "href": "/pts/v2/payments/6938894575296498704951/reversals"
   },
   "self": {
     "method": "GET",
     "href": "/pts/v2/payments/6938894575296498704951"
   },
   "capture": {
     "method": "POST",
     "href": "/pts/v2/payments/6938894575296498704951/captures"
   }
  },
  "clientReferenceInformation": {
   "code": "test123",
   "transactionId": "uniqueValue2"
  },
  "id": "6938894575296498704951",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "9900.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "accountFeatures": {
      "category": "A",
      "group": "0"
    },
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"9F36020015910AB58D60185BEF0247303072179F1804303030318600E04DA9F580903B1BAEDFD1438BA48"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "188589",
    "approvalCode": "831000",
    "networkTransactionId": "016153570198200",
    "retrievalReferenceNumber": "324704188589",
    "transactionId": "016153570198200",
    "responseCode": "00",
    "avs": {
      "code": "2"
    }
  },
  "reconciliationId": "6938894575296498704951",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-09-05T04:50:58Z"
}
```

Authorization with Contactless EMV and Online PIN

For an EMV contactless payment, the customer taps the card on the terminal. The terminal and chip use near-field communication (NFC) to communicate with each other. For an online PIN transaction, the customer uses a PIN to verify their identity and the issuer verifies the PIN.

Online PIN transactions are supported by these card types:

- Visa
- Mastercard
- American Express
- Discover

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for Processing an Authorization with Contactless EMV and Online PIN

clientReferenceInformation.code	
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
merchantInformation.transactionLocalDateTime	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	
pointOfSaleInformation.emv.cardSequenceNumber	
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.encryptedKeySerialNumber	
pointOfSaleInformation.encryptedPin	
pointOfSaleInformation.entryMode	Set the value to contactless for an EMV payment.
pointOfSaleInformation.pinBlockEncodingFormat	
pointOfSaleInformation.terminalCapability	Set the value to 5.
pointOfSaleInformation.terminalPinCapability	
pointOfSaleInformation.trackData	
processingInformation.commerceIndicator	Set the value to retail.

REST Example: Processing an Authorization with Contactless EMV and Online PIN

Request

```
{
  "clientReferenceInformation": {
    "code": "test123",
    "transactionId": "uniqueValue3",
    "partner": {
      "thirdPartyCertificationNumber": "testTPCN"
    }
  },
  "processingInformation": {
    "commerceIndicator": "retail",
    "authorizationOptions": {
      "partialAuthIndicator": "true"
    }
  },
  "paymentInformation": {
     "card": {
        "type": "001"
       }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "9900.00",
      "currency": "USD"
    }
  },
  "pointOfSaleInformation": {
    "entryMode": "contactless",
    "terminalCapability": 4,
     "terminalPinCapability": 6,
    "emv": {
      "tags":
 "5F3401019F3303E0F8C8950580800480009F370465B81A3A9F100706011203A0A0009F2608E9D097D1901E8AB99F360200002
      "cardSequenceNumber": "01"
    },
    "trackData": ";4761xxxxxxxxxxxx=251220111478549?",
    "pinBlockEncodingFormat":0,
    "encryptedPin": "F509429A3C3FD201",
    "encryptedKeySerialNumber": "FFFF1B1D140000200001"
  },
   "merchantInformation": {
  "transactionLocalDateTime": "20230724085022"
 }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6938904668436727104951/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6938904668436727104951"
    },
    "capture": {
```

```
   "method": "POST",
   "href": "/pts/v2/payments/6938904668436727104951/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "test123",
  "transactionId": "uniqueValue3"
 },
 "id": "6938904668436727104951",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "9900.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "accountFeatures": {
   "category": "A",
   "group": "0"
  },
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "emv": {
   "tags":
"9F3602001591 0AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
  }
 },
 "processorInformation": {
  "systemTraceAuditNumber": "188851",
  "approvalCode": "831000",
  "networkTransactionId": "016153570198200",
  "retrievalReferenceNumber": "324705188851",
  "transactionId": "016153570198200",
  "responseCode": "00",
  "avs": {
   "code": "2"
  }
 },
 "reconciliationId": "6938904668436727104951",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-09-05T05:07:47Z"
}
```

Authorization with Magnetic Stripe Swipe

Although EMV chips on payment cards have become common, sometimes the EMV chip cannot be used to validate the cardholder. In these instances, you can choose to validate the cardholder by using the magnetic stripe on back of the payment card.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing an Authorization with Swiped Track Data

clientReferenceInformation.code	
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
merchantInformation.transactionLocalDateTime	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	
pointOfSaleInformation.entryMode	Set the value to `swiped`.
pointOfSaleInformation.terminalCapability	
pointOfSaleInformation.terminalPinCapability	
pointOfSaleInformation.trackData	
processingInformation.commerceIndicator	Set the value to `retail`.

REST Example: Processing an Authorization with Swiped Track Data

Request

```
{
   "clientReferenceInformation": {
      "code": "ABC123",
      "partner": {
         "thirdPartyCertificationNumber": "123456789012"
      }
   },
   "processingInformation": {
      "commerceIndicator": "retail"
      }
   },
  "pointOfSaleInformation": {
    "trackData": ";4111xxxxxxxxxxxx=231220112345678?",
    "entryMode": "swiped",
    "terminalCapability": "4"
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "9601.00",
      "currency": "USD"
    }
  },
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6869553167546562203955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6869553167546562203955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6869553167546562203955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "ABC123"
  },
  "id": "6869553167546562203955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "9601.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "1"
    }
  },
  "reconciliationId": "63427009RIT9HBR9",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-06-16T22:41:57Z"
}
```

Authorizations with Hand-Keyed Data

Under certain circumstances, you might choose to manually enter (hand key) a customer's data to obtain an authorization.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for Processing an Authorization with Hand-Keyed Data

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

clientReferenceInformation.code

clientReferenceInformation.partner.thirdPartyCertificationNumber **Cybersource provides the value for this field.**

clientReferenceInformation.transactionId

merchantInformation.transactionLocalDateTime

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationyear	
paymentInformation.card.number	
pointOfSaleInformation.cardPresent	Set the value to `true`.
pointOfSaleInformation.entryMode	Set the value to `keyed`.
pointOfSaleInformation.terminalCapability	Set the value to `1`, `2`, `3`, `4`, or `5`.
pointOfSaleInformation.terminalPinCapability	
processingInformation.commerceIndicator	Set the value to `retail`.

REST Example: Processing an Authorization with Hand Keyed Data

Request

```
{
   "clientReferenceInformation": {
      "code": "123456",
      "transactionId": "12233445679",
      "partner": {
         "thirdPartyCertificationNumber": "123456789012"
      }
   },
   "processingInformation": {
      "commerceIndicator": "retail",
      "authorizationOptions": {
         "ignoreAvsResult": "true",
         "ignoreCvResult": "true"
      }
   },
   "pointOfSaleInformation": {
      "entryMode": "keyed",
      "terminalCapability": "4",
      "terminalPinCapability": "6"
   },
   "paymentInformation": {
      "card": {
         "number": "4111111111111111",
         "securityCode": "123",
         "expirationMonth": "12",
         "expirationYear": "2031",
         "type": "001"
      }
   },
   "orderInformation": {
      "amountDetails": {
         "totalAmount": "9604.00",
         "currency": "USD"
      },
      "billTo": {
        "postalCode": "94538"
      }
    }
    "merchantInformation": {
      "transactionLocalDateTime": "20230724085022"
    }
}
```

Response to a Successful Request

A successful response returns `status=AUTHORIZED`.

```
{
    "_links": {
      "void": {
        "method": "POST",
        "href": "/pts/v2/payments/6080032225246314603005/voids"
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/6080032225246314603005"
      }
    },
    "clientReferenceInformation": {
      "code": "123456"
    },
    "id": "6080032225246314603005",
    "orderInformation": {
      "amountDetails": {
        "totalAmount": "9604.00",
        "authorizedAmount": "9604.00",
        "currency": "USD"
      }
    },
    "paymentAccountInformation": {
      "card": {
        "type": "001"
      }
    },
    "paymentInformation": {
      "accountFeatures": {
        "category": "A",
        "group": "0"
      },
      "tokenizedCard": {
        "type": "001"
      }
    },
    "processorInformation": {
      "systemTraceAuditNumber": "173156",
      "approvalCode": "831000",
      "cardVerification": {
        "resultCodeRaw": "M",
        "resultCode": "M"
      },
      "networkTransactionId": "016153570198200",
        "transactionId": "016153570198200",
        "responseCode": "00",
        "avs": {
            "code": "Z",
            "codeRaw": "Z"
        }
    },
    "reconciliationId": "6080032225246314603005",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2020-12-15T03:33:42Z"
    }
```

Authorization for Cash Advance with Credit Card

Using the cash advance feature, a cardholder can withdraw cash against their credit card account limit at their bank. The cardholder presents their credit card and identification to bank staff or uses the bank's card terminal to complete this transaction.

> **Important**
>
> The cash advance with credit card at ATM option is not supported currently.

These card types support cash advance with credit card transactions in the U.S:

- Discover. The minimum transaction amount is 10.00.
- Mastercard
- Visa

Fields Specific to This Use Case

processingInformation.authorizationOptions.cashAdvanceIndicator	**Set the value to** `true`**.**
merchantInformation.categoryCode	**Set the value to** `6010`**. This field is not required if merchant category code** `6010` **is configured in the merchant account. If sent, this field overrides the value in the merchant account.**

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorization for Cash Advance with Credit Card

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

clientReferenceInformation.code	
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
merchantInformation.categoryCode	**Set the value to `6010`. This field is not required if merchant category code `6010` is configured in the merchant account. If sent, this field overrides the value in the merchant account.**
merchantInformation.transactionLocalDateTime	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	
pointOfSaleInformation.emv.cardSequenceNumber	
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.encryptedKeySerialNumber	
pointOfSaleInformation.encryptedPin	
pointOfSaleInformation.entryMode	
pointOfSaleInformation.pinBlockEncodingFormat	
pointOfSaleInformation.terminalCapability	
pointOfSaleInformation.terminalPinCapability	
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.cashAdvanceIndicator	**Set the value to `true`.**
processingInformation.commerceIndicator	**Set the value to `retail`.**

Related Information

- *API field reference guide for the REST API*

REST Example: Authorization for Cash Advance with Credit Card

Request

```
{
 "clientReferenceInformation": {
  "code": "Cash Advance",
  "transactionId": "uniqueValue1",
  "partner": {
   "thirdPartyCertificationNumber": "testTPCN"
  }
 },
 "processingInformation": {
  "authorizationOptions": {
    "cashAdvanceIndicator": "true"
  },
  "commerceIndicator": "retail"
 },
 "paymentInformation": {
   "card": {
          "type": "001"
    }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "9900.00",
   "currency": "USD"
  }
 },
 "pointOfSaleInformation": {
  "entryMode": "contact",
  "terminalCapability": 4,
   "terminalPinCapability": 6,
  "emv": {
   "tags":
"5F3401019F3303E0F8C8950580800480009F370465B81A3A9F100706011203A0A0009F2608E9D097D1901E8AB99F36020002
   "cardSequenceNumber": "01"
  },
  "trackData": ";4761xxxxxxxxxxxx=251220111478549?",
  "pinBlockEncodingFormat":0,
  "encryptedPin": "F509429A3C3FD201",
  "encryptedKeySerialNumber": "FFFF1B1D140000200001"
 },
  "merchantInformation": {
 "transactionLocalDateTime": "20230724085022"
}
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6938891699856080004953/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6938891699856080004953"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6938891699856080004953/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "test123",
  "transactionId": "uniqueValue1"
 },
 "id": "6938891699856080004953",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "9900.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "accountFeatures": {
   "category": "A",
   "group": "0"
  },
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "emv": {
   "tags":
"9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
  }
 },
 "processorInformation": {
  "systemTraceAuditNumber": "188535",
  "approvalCode": "831000",
  "networkTransactionId": "016153570198200",
  "retrievalReferenceNumber": "324704188535",
  "transactionId": "016153570198200",
  "responseCode": "00",
  "avs": {
   "code": "2"
```

```
    }
  },
  "reconciliationId": "6938891699856080004953",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-09-05T04:46:10Z"
}
```

Captures

This section provides the information you need in order to capture an authorized transaction.

Supported Card Types

All supported card types can process captures. .

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/captures`
Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures`
The `{id}` is the transaction ID returned in the authorization response.

Required Fields for Capturing an Authorization

Use these required fields for capturing an authorization.

clientReferenceInformation.code	This field value maps from the original authorization, sale, or credit transaction.
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	

Capturing an Authorization Using REST APIs

1. Pass the original authorization ID in the URL, and send the service request to:

   `POST https://<url_prefix>/v2/payments/{id}/captures`
   Use one of these URL prefixes:

   - Test: `apitest.cybersource.com`
   - Production: `api.cybersource.com`
   - Production in India: `api.in.cybersource.com`

   Where `id` is the authorization ID returned in the authorization response.

   ```
   {
     "id": "6481692924466004003001"
   }
   ```

The URL with the `id` value is included in the authorization response:

```
{
 "_links": {
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6481692924466004003001/captures"
  }
 }
```

2. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see *Transaction Response Codes*.

REST Example: Capturing an Authorization

Request

```
{
   "clientReferenceInformation": {
      "code": "ABC123"
   },
   "orderInformation": {
      "amountDetails": {
         "totalAmount": "100.00",
         "currency": "EUR"
   }
}
```

Response to a Successful Request

```
{
   "_links": {
      "void": {
         "method": "POST",
         "href": "/pts/v2/captures/6662994431376681303954/voids"
      },
      "self": {
         "method": "GET",
         "href": "/pts/v2/captures/6662994431376681303954"
      }
   },
   "clientReferenceInformation": {
      "code": "1666299443215"
   },
   "id": "6662994431376681303954",
   "orderInformation": {
      "amountDetails": {
         "totalAmount": "100.00",
         "currency": "EUR"
      }
   },
   "reconciliationId": "66535942B9CGT52U",
   "status": "PENDING",
   "submitTimeUtc": "2022-10-20T20:57:23Z"
}
```

Card Present Connect | Mass Transit Processing

This section shows you how to process card-present transactions for Mass Transit.

Additional Resources for Card Present Connect | Mass Transit

For more information on Card Present Connect | Mass Transit:

- *Card Present Connect | Mass Transit Developer Guide*
- *API Field Reference Using the REST API*
- Github repositories: *https://github.com/Cybersource*

Mass Transit Payment Services Using EMV and Card Data

You can request these payment services for mass transit with EMV and card data:

- Authorization for account verification and debt recovery
- Sale for aggregated fares and debt recovery
- Stand-alone credit

This table shows which EMV tags are:

- M: mandatory
- P: prohibited
- O: optional
- C: conditional (Send the tag when it is present in card and terminal.)

EMV Data Elements and Tags

Data Element	EMV Tag	American Express	Discover PAYG	Mastercard PAYG	Visa MTT
Transaction Date	9A	M	M	M	M
Transaction Type	9C	M	M	M	M
Transaction Currency Code	5F2A	M	M	M	M
Terminal Country Code	9F1A	M	M	M	M
Amount Authorized	9F02	M	M	M	M
Amount Other	9F03	M	M	M	M
Application PAN Sequence Number	5F34	M	P	C	O
Application Transaction Counter (ATC)	9F36	M	M	M	M
Application Interchange Profile (AIP)	82	M	M	M	M
Dedicated File (DF) Name	84	M	M	M	M
Terminal Verification Results (TVR)	95	M	M	M	M
Issuer Application Data	9F10	M	M	M	M
Application Cryptogram	9F26	M	M	M	M
Cryptogram Information Data (CID)	9F27	M	O	M	O
Terminal Capabilities	9F33	M	M	M	M
Cardholder Verification Method (CVM) Results	9F34	O	O	M	O
Unpredictable Number (UN)	9F37	M	M	M	M

Data Element	EMV Tag	American Express	Discover PAYG	Mastercard PAYG	Visa MTT
Form Factor Indicator	9F6E	C*	C	O (Authorizations) P (Refunds)	C
Mastercard Authenticated Application Data	9F60	Does not apply	Does not apply	O	Does not apply
Mastercard Kernel Identifier#Terminal	96	Does not apply	Does not apply	O	Does not apply

*For contactless American Express transactions, if the form factor indicator data is available on the card, the merchant, acquirer, or processor must forward this information to the issuer.

American Express Account Status Check Authorization with EMV Data

This section describes how to process an American Express account status check authorization with EMV data for a nominal amount of 1.00 USD or more. The required function code is 190.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for a American Express Account Status Check AVR Authorization with EMV Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	**Set this field to** `TransitDA BAU nominal value auth`**.**
clientReferenceInformation.partner.solutionId	**Cybersource provides the value for this field.**
clientReferenceInformation.partner.thirdPartyCertificationNumber	**Cybersource provides the value for this field.**
clientReferenceInformation.transactionId	

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	Set this field to 003.
pointOfSaleInformation.catLevel	Set this field to 2.
pointOfSaleInformation.emv.cardSequenceNumber	Set this field to 00.
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set this field to contactless.
pointOfSaleInformation.terminalCapability	Set this field to 5.
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to 0.
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.aggregatedAuthIndicator	Set this field to true.
processingInformation.authorizationOptions.deferredAuthIndicator	Set this field to true.
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set this field to retail.
processingInformation.industryDataType	Set this field to transit.

Related information

- API Field Reference for the REST API

REST Example: American Express Account Status Check Authorization with EMV Data

Request

```
{
  "orderInformation": {
    "amountDetails": {
      "currency": "EUR",
      "totalAmount": "3.00"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "003"
    }
  },
  "processingInformation": {
    "capture": false,
    "captureOptions": {
      "dateToCapture": "0901"
    },
    "industryDataType": "transit",
    "commerceIndicator": "retail",
    "authorizationOptions": {
      "partialAuthIndicator": false,
      "deferredAuthIndicator": true,
      "aggregatedAuthIndicator": true
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"9A032309019C01005F2A0209789F1A0203809F02060000000000009F03060000000000009F36020002820219C08408A00
      "cardSequenceNumber": "00"
    },
    "catLevel": "2",
    "entryMode": "contactless",
    "trackData": ";374245XXXXXXXXXX=241270115041234500000?",
    "terminalId": "12345678",
    "terminalCapability": "5",
    "terminalPinCapability": "0"
  },
  "clientReferenceInformation": {
    "comments": "TransitDA BAU nominal value auth",
    "code": "v7qWAImW6e",
    "partner": {
      "solutionId": "BUALWMZK",
      "thirdPartyCertificationNumber": "condue211609"
    },
    "transactionId": "Fg1xkLJGMmmmvwbB9qWAImW6e"
  }
}
```

Response to a Successful Request

```
  {
    "_links":#{
      "authReversal":#{
        "method":#"POST",
        "href":#"/pts/v2/payments/6984001952686181104951/reversals"
      },
      "self":#{
        "method":#"GET",
        "href":#"/pts/v2/payments/6984001952686181104951"
      },
      "capture":#{
        "method":#"POST",
        "href":#"/pts/v2/payments/6984001952686181104951/captures"
      }
    },
    "clientReferenceInformation":#{
      "code":#"v7qWAImW6e",
      "partner":#{
        "solutionId":#"BUALWMZK"
      },
      "transactionId":#"Fg1xkLJGMmmmvwbB9qWAImW6e"
    },
    "id":#"6984001952686181104951",
    "orderInformation":#{
      "amountDetails":#{
        "authorizedAmount":#"3.00",
        "currency":#"EUR"
      }
    },
    "paymentAccountInformation":#{
      "card":#{
        "type":#"003"
      }
    },
    "paymentInformation":#{
      "accountFeatures":#{
        "category":#"AX",
        "group":#"0"
      },
      "tokenizedCard":#{
        "type":#"003"
      },
      "card":#{
        "type":#"003"
      }
    },
    "pointOfSaleInformation":#{
      "emv":#{
        "tags":#"9F2701809F34031F02025F340100"
      }
    },
    "processorInformation":#{
      "systemTraceAuditNumber":#"037806",
      #"approvalCode":#"845614",
      "networktransactionId":#"001032401292273",
      "retrievalReferenceNumber":#"330009037806",
      "transactionId":#"001032401292273",
      "responseCode":#"00",
      "avs":#{
        "code":#"2"
      }
    },
    "reconciliationId":#"6984001952686181104951",
    "status":#"AUTHORIZED",
    "submitTimeUtc":#"2023-10-27T09:49:56Z"
  }
```

American Express Delayed Online Authorization with EMV Data

This section describes how to process an American Express delayed online authorization with EMV data for a nominal amount of 1.00 USD or more. The required function code is 100.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for a American Express Delayed Online Authorization with EMV Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	Set this field to TransitDA BAU nominal value auth.
clientReferenceInformation.partner.solutionId	Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	Set this field to 003.
pointOfSaleInformation.catLevel	Set this field to 2.
pointOfSaleInformation.emv.cardSequenceNumber	Set this field to 00.
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set this field to contactless.
pointOfSaleInformation.terminalCapability	Set this field to 5.
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to 0.
pointOfSaleInformation.trackData	
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set this field to retail.
processingInformation.industryDataType	Set this field to transit.

Related information

- API Field Reference for the REST API

REST Example: American Express Delayed Online Authorization with EMV Data

Request

```
{
  "orderInformation": {
    "amountDetails": {
      "currency": "EUR",
      "totalAmount": "8.00"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "003"
    }
  },
  "processingInformation": {
    "captureOptions": {
      "dateToCapture": "0901"
    },
    "industryDataType": "transit",
    "commerceIndicator": "retail"
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags":
 "9A032309019C01005F2A0209789F1A0203809F02060000000000009F03060000000000009F36020002820219C08408A00
      "cardSequenceNumber": "00"
    },
    "catLevel": "2",
    "entryMode": "contactless",
    "trackData": ";341111XXXXXXXXXX=241270215041234500000?",
    "terminalId": "12345678",
    "terminalCapability": "5",
    "terminalPinCapability": "0"
  },
  "clientReferenceInformation": {
    "comments": "TransitDA BAU full value auth",
    "code": "v7qWAImW6e",
    "partner": {
      "solutionId": "BUALWMZK",
      "thirdPartyCertificationNumber": "condue211609"
    },
    "transactionId": "Fg1xkLJGMmmmvwbB9qWAImW6e"
  }
}
```

Response to a Successful Request

```
{
  "_links":#{
    "authReversal":#{
      "method":#"POST",
   #"href":#"/pts/v2/payments/6984003567376178404953/reversals"
  },
  #"self":#{
   "method":#"GET",
   "href":#"/pts/v2/payments/6984003567376178404953"
  },
    "capture":#{
   "method":#"POST",
      "href":#"/pts/v2/payments/6984003567376178404953/captures"
    }
  },
  "clientReferenceInformation":#{
    "code":#"v7qWAImW6e",
    "partner":#{
      "solutionId":#"BUALWMZK"
    },
  "transactionId":#"Fs8xkLJGNslmvwbZ9qWAImW6e"
  },
  "id":#"6984003567376178404953",
  "orderInformation":#{
    "amountDetails":#{
      "authorizedAmount":#"8.00",
      "currency":#"EUR"
    }
  },
  "paymentAccountInformation":#{
    "card":#{
      "type":#"003"
    }
  },
  "paymentInformation":#{
    "accountFeatures":#{
      "category":#"AX",
      "group":#"0"
    },
    "tokenizedCard":#{
      "type":#"003"
    },
    "card":#{
      "type":#"003"
    }
  },
  "pointOfSaleInformation":#{
    "emv":#{
      "tags":#"9F2701809F34033F00005F340100910AEE43F0FD6F46AABF3030"
    }
  },
  "processorInformation":#{
    "systemTraceAuditNumber":#"037809",
    "approvalCode":#"437964",
    "networktransactionId":#"000002605437964",
    "retrievalReferenceNumber":#"330009037809",
    "transactionId":#"000002605437964",
    "responseCode":#"00",
    "avs":#{
      "code":#"2"
    }
  },
  "reconciliationId":#"69840035673761784O4953",
  "status":#"AUTHORIZED",
  "submitTimeUtc":#"2023-10-27T09:52:39Z"
}
```

Discover Authorization with EMV Data

A Discover authorization with EMV data is an authorization request that can be for a nominal amount of 1.00 USD or a fare amount up to 15.00 USD. Mass transit Discover transactions are supported only in the U.S.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for a Discover Authorization with EMV Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	Set this field to `TransitDA BAU nominal value auth`.
clientReferenceInformation.partner.solutionId	Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	Set this field to `1.00`.
paymentInformation.card.type	Set this field to `004`.
pointOfSaleInformation.catLevel	Set this field to `2`.
pointOfSaleInformation.emv.cardSequenceNumber	Set this field to `99`.
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set this field to `contactless`.
pointOfSaleInformation.terminalCapability	Set this field to `5`.
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to `0`.

pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.aggregatedAuthIndicator	Set this field to `true`.
processingInformation.authorizationOptions.deferredAuthIndicator	Set this field to `true`.
processingInformation.commerceIndicator	Set this field to `retail`.
processingInformation.industryDataType	Set this field to `transit`.

Related information

- API Field Reference for the REST API

REST Example: Discover Authorization with EMV Data

Request

```
{
   "clientReferenceInformation": {
      "comments": "TransitDA BAU nominal value auth",
      "code": "123456",
      "transactionId": "1346334405",
      "partner": {
         "solutionId": "123456",
         "thirdPartyCertificationNumber": "123456789012"
      }
   },
   "processingInformation": {
      "industryDataType": "transit",
      "capture": "false",
      "commerceIndicator": "retail",
      "authorizationOptions": {
         "deferredAuthIndicator": "true",
         "aggregatedAuthIndicator": "true"
      }
   },
   "orderInformation": {
      "amountDetails": {
         "totalAmount": "1.00",
         "currency": "USD"
      }
   },
   "paymentInformation": {
      "card": {
         "type": "004"
      }
   },
   "pointOfSaleInformation": {
      "terminalId": "12345678",
      "catLevel": "2",
      "entryMode": "contactless",
      "terminalCapability": "5",
      "terminalPinCapability": "0",
      "emv": {
        "tags":
 "9F2608101F3F75E8596414820211009F360200019F2701409F100A0115100000000000000000950500000000009F370438A87
        "cardSequenceNumber": "99"
      },
      "trackData": ";651000XXXXXXXXXX=4912201180408850000000?"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6920241974736435904951/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6920241974736435904951"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6920241974736435904951/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "123456",
    "comments": "TransitDA BAU nominal value auth",
    "partner": {
      "solutionId": "123456"
    },
    "transactionId": "1346334405"
  },
  "id": "6920241974736435904951",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "1.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "004"
    }
  },
  "paymentInformation": {
    "accountFeatures": {
      "category": "DI",
      "group": "0"
    },
    "tokenizedCard": {
      "type": "004"
    },
    "card": {
      "type": "004"
    }
    },
    "pointOfSaleInformation": {
        "emv": {
            "tags": "9F2701409F3501259F36020001"
        }
    },
    "processorInformation": {
        "systemTraceAuditNumber": "033732",
        "approvalCode": "813783",
        "cardReferenceData": "05",
        "networktransactionId": "VISJ     303226529970011",
        "retrievalReferenceNumber": "322614033732",
        "consumerAuthenticationResponse": {
            "code": "0",
            "codeRaw": "0"
        },
        "transactionId": "VISJ     303226529970011",
        "responseCode": "00",
        "avs": {
            "code": "2"
        }
    },
    "reconciliationId": "69202419747364359O4951",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2023-08-14T14:43:19Z"
}
```

Mastercard Authorization with EMV Data

This section describes how to process a Mastercard authorization with EMV data for a nominal amount.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for a Mastercard Authorization with EMV Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	Set this field to TransitDA BAU nominal value auth.
clientReferenceInformation.partner.solutionId	Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.

clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	Set this field to 002.
paymentInformation.initiationChannel	
pointOfSaleInformation.catLevel	Set this field to 2.
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set this field to contactless.
pointOfSaleInformation.serviceCode	
pointOfSaleInformation.terminalCapability	Set this field to 5.
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to 0.
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.aggregatedAuthIndicator	Set this field to true.
processingInformation.authorizationOptions.authIndicator	Set this field to 0.
processingInformation.authorizationOptions.deferredAuthIndicator	Set this field to true.
processingInformation.authorizationOptions.transportationMode	
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set this field to retail.
processingInformation.industryDataType	Set this field to transit.

Related information

- API Field Reference for the REST API

REST Example: Mastercard Authorization with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA BAU nominal value auth",
    "code": "10000568",
    "transactionId": "20000568",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "commerceIndicator": "retail",
    "capture": "false",
    "captureOptions": {
      "dateToCapture": "0425"
    },
    "authorizationOptions": {
      "authIndicator": "0",
      "deferredAuthIndicator": "true",
      "aggregatedAuthIndicator": "true",
      "transportationMode": "00"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "002"
    },
    "initiationChannel": "00"
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",
    "entryMode": "contactless",
    "terminalCapability": "5",
    "terminalPinCapability": "0",
    "emv": {
      "tags":
"5F2A0209768407A00000000410109F360200039F03060000000000009C01005F3401019F10120110A0000F04000000000
    },
    "trackData": ";5413XXXXXXXXXXXX=49122010123456789?",
    "serviceCode": "201"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508877845426512004004"
    }
  },
  "clientReferenceInformation": {
    "code": "10000574",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000574"
  },
  "errorInformation": {
      "reason": "AUTH_DECLINE_CAPTURE_POSSIBLE",
      "message": "Authorization Declined. Follow-on Capture can be processed."
    },
    "id": "6508877845426512004004",
    "pointOfSaleInformation": {
      "emv": {
        "tags":
 "9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
      }
    },
    "processorInformation": {
      "systemTraceAuditNumber": "164207",
      "networktransactionId": "016153570198200",
      "retrievalReferenceNumber": "211511164207",
      "transactionId": "016153570198200",
      "responseCode": "05",
      "avs": {
        "code": "2"
      }
    },
    "status": "AUTHORIZED"
}
```

Visa Account Verification Request (AVR) with EMV Data

This section describes how to process a Visa account verification request (AVR) with EMV data for a zero amount.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for a Visa AVR Authorization with EMV Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	**Set this field to `TransitDA BAU zero value auth`.**
clientReferenceInformation.partner.solutionId	**Cybersource provides the value for this field.**
clientReferenceInformation.partner.thirdPartyCertificationNumber	**Cybersource provides the value for this field.**
clientReferenceInformation.transactionId	

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	**Set this field to `0.00`.**
paymentInformation.card.type	**Set this field to `001`.**
pointOfSaleInformation.catLevel	**Set this field to `2`.**
pointOfSaleInformation.emv.cardSequenceNumber	
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	**Set this field to `contactless`.**
pointOfSaleInformation.terminalCapability	**Set this field to `5`.**
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	**Set this field to `0`.**
pointOfSaleInformation.trackData	
processingInformation.commerceIndicator	**Set this field to `retail`.**

Related information

- API Field Reference for the REST API

REST Example: Visa AVR Authorization with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA BAU zero value auth",
    "code": "10000564",
    "transactionId": "20000564",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "capture": "false",
    "commerceIndicator": "retail"
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "0.00",
      "currency": "EUR"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
      "terminalId": "12345678",
      "catLevel": "2",
      "entryMode": "contactless",
      "terminalCapability": "5",
      "terminalPinCapability": "0",
      "emv": {
         "tags":
 "5F2A0209768407A00000000031010 9F360200029F0306000000000000009C01005F3401019F10201F220100A000000000000
         "cardSequenceNumber": "1"
      },
      "trackData": ";476173XXXXXXXXXX=241220119058254?"
   }
}
```

Response to a Successful Request

```
{
   "_links": {
      "authReversal": {
         "method": "POST",
         "href": "/pts/v2/payments/6508875466126538104002/reversals"
      },
      "self": {
         "method": "GET",
         "href": "/pts/v2/payments/6508875466126538104002"
      },
      "capture": {
         "method": "POST",
         "href": "/pts/v2/payments/6508875466126538104002/captures"
      }
   },
   "clientReferenceInformation": {
      "code": "10000564",
      "partner": {
         "solutionId": "548UHQ8Z"
      },
      "transactionId": "20000564"
   },
   "id": "6508875466126538104002",
   "orderInformation": {
      "amountDetails": {
         "authorizedAmount": "0.00",
         "currency": "EUR"
      }
   },
   "paymentAccountInformation": {
      "card": {
         "type": "001"
      }
   },
   "paymentInformation": {
      "tokenizedCard": {
         "type": "001"
      },
      "card": {
         "type": "001"
```

```
        }
    },
    "processorInformation": {
        "systemTraceAuditNumber": "162930",
        "approvalCode": "831000",
        "merchantAdvice": {
            "code": "01",
            "codeRaw": "M001"
        },
        "responseDetails": "ABC",
        "networktransactionId": "016153570198200",
        "retrievalReferenceNumber": "211511162930",
        "consumerAuthenticationResponse": {
            "code": "2",
            "codeRaw": "2"
        },
        "transactionId": "016153570198200",
        "responseCode": "00",
        "avs": {
            "code": "Y",
            "codeRaw": "Y"
        }
    },
    "reconciliationId": "6508875466126538104002",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2022-04-25T11:52:26Z"
}
```

Response to a Declined Request

```
{
    "_links": {
        "self": {
            "method": "GET",
            "href": "/pts/v2/payments/6508876049646556304003"
        }
    },
    "clientReferenceInformation": {
        "code": "10000566",
        "partner": {
            "solutionId": "548UHQ8Z"
        },
        "transactionId": "20000566"
    },
    "errorInformation": {
        "reason": "AUTH_DECLINE_CAPTURE_POSSIBLE",
        "message": "Authorization Declined. Follow-on Capture can be processed."
    },
    "id": "6508876049646556304003",
    "pointOfSaleInformation": {
        "emv": {
            "tags":
 "9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
        }
    },
    "processorInformation": {
        "systemTraceAuditNumber": "162936",
        "networktransactionId": "016153570198200",
        "retrievalReferenceNumber": "211511162936",
        "transactionId": "016153570198200",
        "responseCode": "05",
        "avs": {
            "code": "2"
        }
    },
    "status": "AUTHORIZED"
}
```

Discover Sale with EMV Data

A sale transaction comprises an authorization and capture. When the fare is more than 15.00 USD, request a sale with EMV data.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for a Discover Sale with EMV Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	Set this field to TransitDA BAU nominal value sale.
clientReferenceInformation.partner.solutionId	Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	Set this field to 1.00.
paymentInformation.card.type	Set this field to 004.
pointOfSaleInformation.catLevel	Set this field to 2.
pointOfSaleInformation.emv.cardSequenceNumber	Set this field to 99.
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set this field to contactless.
pointOfSaleInformation.terminalCapability	Set this field to 5.

pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to `0`.
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.aggregatedAuthIndicator	Set this field to `true`.
processingInformation.authorizationOptions.deferredAuthIndicator	Set this field to `true`.
processingInformation.capture	Set the value to `true`.
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set this field to `retail`.
processingInformation.industryDataType	Set this field to `transit`.

Related information

- API Field Reference for the REST API

REST Example: Discover Sale with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA BAU full value sale",
    "code": "123456",
    "transactionId": "1357334401",
    "partner": {
      "solutionId": "123456",
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "reconciliationId": "123456789",
    "captureOptions": {
      "dateToCapture": "0818"
    },
    "capture": "true",
    "commerceIndicator": "retail"
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "25.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "004"
    }
  },
  "pointOfSaleInformation": {
      "terminalId": "12345678",
      "catLevel": "2",
      "entryMode": "contactless",
      "terminalCapability": "5",
      "terminalPinCapability": "0",
      "emv": {
        "tags":
 "9F2608101F3F75E8596414820211009F360200019F2701409F100A01151000000000000000950500000000009F370438A87
        "cardSequenceNumber": "99"
      },
      "trackData": ";651000XXXXXXXXXX=49122011804088500000?"
   }
}
```

Response to a Successful Request

```
{
   "_links": {
      "void": {
        "method": "POST",
        "href": "/pts/v2/payments/6920243246666458104951/voids"
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/6920243246666458104951"
      }
   },
   "clientReferenceInformation": {
      "code": "123456",
      "comments": "TransitDA BAU full value sale",
      "partner": {
        "solutionId": "123456"
      },
      "transactionId": "1357334401"
   },
   "id": "6920243246666458104951",
   "orderInformation": {
      "amountDetails": {
        "totalAmount": "15.00",
        "authorizedAmount": "15.00",
        "currency": "USD"
      }
   },
   "paymentAccountInformation": {
      "card": {
        "type": "004"
      }
   },
   "paymentInformation": {
      "accountFeatures": {
        "category": "DI",
        "group": "0"
      },
      "tokenizedCard": {
        "type": "004"
      },
      "card": {
        "type": "004"
      }
   },
   "pointOfSaleInformation": {
      "emv": {
        "tags": "9F2701409F3501259F36020001"
      }
   },
   "processorInformation": {
      "systemTraceAuditNumber": "033735",
      "approvalCode": "378857",
      "cardReferenceData": "05",
      "networktransactionId": "VISJ     303226531251404",
      "retrievalReferenceNumber": "322614033735",
      "consumerAuthenticationResponse": {
        "code": "0",
        "codeRaw": "0"
      },
      "transactionId": "VISJ     303226531251404",
      "responseCode": "00",
      "avs": {
        "code": "2"
      }
   },
   "reconciliationId": "69202432466664581049 51",
   "status": "AUTHORIZED",
   "submitTimeUtc": "2023-08-14T14:45:26Z"
}
```

Visa Deferred Sale with EMV Data

This section describes how to process a deferred sale transaction at the end of the travel period for an aggregated payment.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for a Visa Deferred Sale with EMV Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	**Set the value to** TransitDA BAU full value sale**.**
clientReferenceInformation.partner.solutionId	**Cybersource provides the value for this field.**

clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	Set the value to `001`.
pointOfSaleInformation.catLevel	Set the value to `2`.
pointOfSaleInformation.emv.cardSequenceNumber	
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set the value to `contactless`.
pointOfSaleInformation.terminalCapability	Set the value to `5`.
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set the value to `0`.
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.aggregatedAuthIndicator	Set the value to `true`.
processingInformation.authorizationOptions.deferredAuthIndicator	Set the value to `true`.
processingInformation.capture	Set the value to `true`.
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set the value to `retail`.
processingInformation.industryDataType	Set the value to `transit`.

Related information

- API Field Reference for the REST API

REST Example: Visa Deferred Sale with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA BAU full value sale",
    "code": "10000565",
    "transactionId": "20000565",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "processingInformation.commerceIndicator": "retail",
    "capture": "true",
    "captureOptions": {
      "dateToCapture": "0425"
    },
    "authorizationOptions": {
      "deferredAuthIndicator": "true",
      "aggregatedAuthIndicator": "true"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",
    "entryMode": "contactless",
    "terminalCapability": "5",
    "terminalPinCapability": "0",
    "emv": {
      "tags":
 "5F2A0209768407A00000000310109F360200029F03060000000000009C01005F3401019F10201F220100A000000000000
      "cardSequenceNumber": "1"
    },
    "trackData": ";4761XXXXXXXXXXXX=241220119058254?"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/payments/6508875814676551204001/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508875814676551204001"
    }
  },
  "clientReferenceInformation": {
    "code": "10000565",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000565"
  },
  "id": "6508875814676551204001",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "authorizedAmount": "10.00",
      "currency": "EUR"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "accountFeatures": {
      "group": "0"
    },
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags":
 "9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "164186",
    "approvalCode": "831000",
    "networktransactionId": "016153570198200",
    "retrievalReferenceNumber": "211511164186",
    "transactionId": "016153570198200",
    "responseCode": "00",
    "avs": {
      "code": "2"
    }
  },
  "reconciliationId": "6508875814676551204001",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-04-25T11:53:01Z"
}
```

Response to a Declined Request with First Ride Protection

```
{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508878333386555704002"
    }
  },
  "clientReferenceInformation": {
    "code": "10000576",
      "partner": {
         "solutionId": "548UHQ8Z"
      },
      "transactionId": "20000576"
   },
   "errorInformation": {
      "reason": "AUTH_DECLINE_CAPTURE_POSSIBLE",
      "message": "Authorization Declined. Follow-on Capture can be processed."
   },
   "id": "6508878333865557040002",
   "pointOfSaleInformation": {
      "emv": {
         "tags":
 "9F3602001591 0AB58D60185BEF0247303072179F1804303031860E04DA9F580903B1BAEDFD1438BA48"
      }
   },
   "processorInformation": {
      "systemTraceAuditNumber": "164212",
      "networktransactionId": "016153570198200",
      "retrievalReferenceNumber": "211511164212",
      "transactionId": "016153570198200",
      "responseCode": "05",
      "avs": {
         "code": "2"
      }
   },
   "status": "DECLINED"
}
```

Tap-Initiated Authorization for Debt Recovery with EMV Data

This section describes how to process a tap-initiated authorization for debt recovery. When a cardholder attempts to use a blocked card at the transit reader, create a new debt recovery authorization request using the chip data from the new tap, along with the fare amount of the previous declined authorization.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for a Tap-Initiated Authorization for Debt Recovery with EMV Data Using the REST API

clientReferenceInformation.code

clientReferenceInformation.comments

Set this field to TransitDA Debt recovery tap auth**.**

clientReferenceInformation.partner.solutionId	Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	
pointOfSaleInformation.catLevel	Set this field to `2`.
pointOfSaleInformation.emv.cardSequenceNumber	Set this field to `1`.
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set this field to `contactless`.
pointOfSaleInformation.terminalCapability	Set this field to `5`.
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to `0`.
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.debtRecoveryIndicator	Set this field to `true`.
processingInformation.authorizationOptions.deferredAuthIndicator	Set this field to `true`.
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set this field to `retail`.
processingInformation.industryDataType	Set this field to `transit`.

Related information

- API Field Reference for the REST API

REST Example: Tap-Initiated Authorization for Debt Recovery with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA Debt recovery tap auth",
    "code": "10000597",
    "transactionId": "20000597",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "captureOptions": {
      "dateToCapture": "0425"
    },
    "commerceIndicator": "retail",
    "authorizationOptions": {
      "debtRecoveryIndicator": "true",
      "deferredAuthIndicator": "true"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",
    "entryMode": "contactless",
    "terminalCapability": "5",
    "terminalPinCapability": "0",
    "emv": {
      "tags":
 "5F2A0209768407A0000000031010 9F360200029F0306000000000000009C01005F3401019F10201F220100A000000000000
      "cardSequenceNumber": "1"
    },
    "trackData": ";476173XXXXXXXXXX=241220119058254?"
  }
}
```

Response to a Declined Request

```
{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508883585936636904003"
    }
  },
  "clientReferenceInformation": {
    "code": "10000597",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000597"
  },
  "errorInformation": {
    "reason": "PROCESSOR_DECLINED",
    "message": "Decline - General decline of the card. No other information provided by the issuing bank."
```

```
   },
   "id": "6508883585936636904003",
   "pointOfSaleInformation": {
      "emv": {
         "tags":
 "9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
      }
   },
   "processorInformation": {
      "systemTraceAuditNumber": "163648",
      "networktransactionId": "016153570198200",
      "retrievalReferenceNumber": "211512163648",
      "transactionId": "016153570198200",
      "responseCode": "05",
      "avs": {
         "code": "2"
      }
   },
   "status": "DECLINED"
}
```

Merchant-Initiated Authorizations for Debt Recovery with Stored Card Data

This section describes how to process a merchant-initiated authorization for debt recovery with stored card data.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for a Merchant-Initiated Authorization for Debt Recovery with Stored Card Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	**Set this field to** TransitDA Debt recovery MIT auth**.**
clientReferenceInformation.partner.solutionId	**Cybersource provides the value for this field.**
clientReferenceInformation.partner.thirdPartyCertificationNumber	**Cybersource provides the value for this field.**
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	

orderInformation.amountDetails.totalAmount	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
paymentInformation.card.type	
processingInformation.authorizationOptions.debtRecoveryIndicator	Set this field to true.
processingInformation.authorizationOptions.ignoreAvsResult	Set this field to true.
processingInformation.authorizationOptions.ignoreCvResult	Set this field to true.
processingInformation.authorizationOptions.initiator.credentialStoredOnFile	Set this field to false.
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionId	
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	Set this field to 1.
processingInformation.authorizationOptions.initiator.storedCredentialUsed	Set this field to true.
processingInformation.authorizationOptions.initiator.type	Set this field to merchant.
processingInformation.commerceIndicator	Set this field to moto.
processingInformation.industryDataType	Set this field to transit.

Related information

- API Field Reference for the REST API

REST Example: Merchant-Initiated Authorization for Debt Recovery with Stored Card Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA Debt recovery MIT auth",
    "code": "10000596",
    "transactionId": "20000596",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "commerceIndicator": "moto",
    "industryDataType": "transit",
    "authorizationOptions": {
      "debtRecoveryIndicator": "true",
      "ignoreAvsResult": "true",
      "ignoreCvResult": "true",
      "initiator": {
        "type": "merchant",
        "credentialStoredOnFile": "false",
        "storedCredentialUsed": "true",
        "merchantInitiatedTransaction": {
          "reason": "1",
          "previousTransactionId": "016153570198200"
        }
      }
    }
  },
  "paymentInformation": {
    "card": {
      "number": "476173XXXXXXXXXX",
      "expirationMonth": "12",
      "expirationYear": "2024",
      "type": "001"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  }
}
```

Response to a Declined Request

```
{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508883374816631904001"
    }
  },
  "clientReferenceInformation": {
    "code": "10000596",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000596"
  },
  "errorInformation": {
    "reason": "PROCESSOR_DECLINED",
    "message": "Decline - General decline of the card. No other information provided by the issuing bank."
  },
  "id": "6508883374816631904001",
  "pointOfSaleInformation": {
    "emv": {
      "tags":
 "9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "164869",
    "networktransactionId": "016153570198200",
    "retrievalReferenceNumber": "211512164869",
    "transactionId": "016153570198200",
    "responseCode": "05",
    "avs": {
      "code": "1"
    }
  },
  "status": "DECLINED"
}
```

Tap-Initiated Sale for Mastercard Debt Recovery with EMV Data

This section describes how to process a tap-initiated sale for Mastercard debt recovery When a cardholder attempts to use a blocked card at the transit reader, create a new debt recovery sale request using the chip data from the new tap, along with the fare amount of the previous declined authorization.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for a Tap-Initiated Sale for Mastercard Debt Recovery with EMV Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	Set this field to `TransitDA Debt recovery tap sale`.
clientReferenceInformation.partner.solutionId	Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	
paymentInformation.initiationChannel	Set this field to `00`.
pointOfSaleInformation.catLevel	Set this field to `2`.
pointOfSaleInformation.emv.tags	

pointOfSaleInformation.entryMode	Set this field to `contactless`.
pointOfSaleInformation.serviceCode	
pointOfSaleInformation.terminalCapability	Set this field to `5`.
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to `0`.
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.authIndicator	Set this field to `1`.
processingInformation.authorizationOptions.debtRecoveryIndicator	Set this field to `true`.
processingInformation.authorizationOptions.transportationMode	
processingInformation.capture	Set this field to `true`.
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set this field to `retail`.
processingInformation.industryDataType	Set this field to `transit`.

Related information

- API Field Reference for the REST API

REST Example: Tap-Initiated Sale for Mastercard Debt Recovery with EMV Data

Request

```
{
   "clientReferenceInformation": {
      "comments": "TransitDA Debt recovery tap sale",
      "code": "10000575MC",
      "transactionId": "20000575MC",
      "partner": {
         "thirdPartyCertificationNumber": "BPCDRC220403",
         "solutionId": "548UHQ8Z"
      }
   },
   "processingInformation": {
      "industryDataType": "transit",
      "commerceIndicator": "retail",
      "capture": "true",
      "captureOptions": {
         "dateToCapture": "0425"
      },
      "authorizationOptions": {
         "authIndicator": "1",
         "debtRecoveryIndicator": "true",
         "transportationMode": "00"
      }
   },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "002"
    },
    "initiationChannel": "00"
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",
    "entryMode": "contactless",
    "terminalCapability": "5",
    "terminalPinCapability": "0",
    "emv": {
      "tags":
 "5F2A0209768407A00000000041010 9F360200039F03060000000000009C01005F3401019F10120110A0000F04000000000
    },
    "trackData": ";5413XXXXXXXXXXXX=49122010123456789?",
    "serviceCode": "201"
  }
}
```

Response to a Declined Request

```
{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508878027476513004000"
    }
  },
  "clientReferenceInformation": {
    "code": "10000575MC",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000575MC"
  },
  "errorInformation": {
    "reason": "PROCESSOR_DECLINED",
    "message": "Decline - General decline of the card. No other information provided by the issuing bank."
  },
  "id": "6508878027476513004000",
  "pointOfSaleInformation": {
    "emv": {
      "tags":
 "9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "162956",
    "networktransactionId": "016153570198200",
    "retrievalReferenceNumber": "211511162956",
    "transactionId": "016153570198200",
    "responseCode": "05",
    "avs": {
      "code": "2"
    }
  },
  "status": "DECLINED"
}
```

Tap-Initiated Sale for Visa Debt Recovery with EMV Data

This section describes how to process a tap-initiated sale for Visa debt recovery. When a cardholder attempts to use a blocked card at the transit reader, create a new debt recovery sale request using the chip data from the new tap, along with the fare amount of the previous declined authorization.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for a Tap-Initiated Sale for Visa Debt Recovery with EMV Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	Set this field to `TransitDA Debt recovery tap sale`.
clientReferenceInformation.partner.solutionId	Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	
paymentInformation.initiationChannel	Set this field to `00`.
pointOfSaleInformation.catLevel	Set this field to `2`.
pointOfSaleInformation.emv.tags	

pointOfSaleInformation.entryMode	Set this field to `contactless`.
pointOfSaleInformation.serviceCode	
pointOfSaleInformation.terminalCapability	Set this field to `5`.
pointOfSaleInformation.terminalId	pointOfSaleInformation.terminalId
pointOfSaleInformation.terminalPinCapability	Set this field to `0`.
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.authIndicator	Set this field to `1`.
processingInformation.authorizationOptions.debtRecoveryIndicator	Set this field to `true`.
processingInformation.authorizationOptions.deferredAuthIndicator	Set this field to `true`.
processingInformation.authorizationOptions.transportationMode	
processingInformation.capture	Set this field to `true`.
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set this field to `retail`.
processingInformation.industryDataType	Set this field to `transit`.

Related information

- API Field Reference for the REST API

REST Example: Tap-Initiated Sale for Visa Debt Recovery with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA Debt recovery tap sale",
    "code": "10000575",
    "transactionId": "20000575",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "commerceIndicator": "retail",
    "capture": "true",
    "captureOptions": {
      "dateToCapture": "0425"
    },
    "authorizationOptions": {
      "authIndicator": "1",
      "debtRecoveryIndicator": "true",
      "deferredAuthIndicator": "true",
      "transportationMode": "00"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    },
    "initiationChannel": "00"
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",
    "entryMode": "contactless",
    "terminalCapability": "5",
    "terminalPinCapability": "0",
    "emv": {
      "tags":
"5F2A0209768407A000000000410109F360200039F03060000000000009C01005F3401019F10120110A0000F040000000000
    },
    "trackData": ";4413XXXXXXXXXXXX=49122010123456789?",
    "serviceCode": "201"
  }
}
```

Response to a Declined Request

```
{
    "_links": {
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/6508878027476513004004"
      }
    },
    "clientReferenceInformation": {
      "code": "10000575",
      "partner": {
        "solutionId": "548UHQ8Z"
      },
      "transactionId": "20000575"
    },
    "errorInformation": {
      "reason": "PROCESSOR_DECLINED",
      "message": "Decline - General decline of the card. No other information provided by the issuing bank."
    },
    "id": "6508878027476513004004",
    "pointOfSaleInformation": {
      "emv": {
        "tags":
  "9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
      }
    },
    "processorInformation": {
      "systemTraceAuditNumber": "162956",
      "networktransactionId": "016153570198200",
      "retrievalReferenceNumber": "211511162956",
      "transactionId": "016153570198200",
      "responseCode": "05",
      "avs": {
        "code": "2"
      }
    },
    "status": "DECLINED"
}
```

Merchant-Initiated Sale for Debt Recovery with Stored Card Data

This section describes how to process a bundled authorization and capture to perform a merchant-initiated sale for debt recovery.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for a Merchant-Initiated Sale for Debt Recovery with Stored Card Data Using the REST API

clientReferenceInformation.code	
clientReferenceInformation.comments	**Set this field to** `TransitDA Debt recovery MIT sale`**.**
clientReferenceInformation.partner.solutionId	**Cybersource provides the value for this field.**
clientReferenceInformation.partner.thirdPartyCertificationNumber	**Cybersource provides the value for this field.**
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
paymentInformation.card.type	

processingInformation.authorizationOptions.authIndicator	Set this field to 1.
processingInformation.authorizationOptions.debtRecoveryIndicator	Set this field to true.
processingInformation.authorizationOptions.ignoreAvsResult	Set this field to true.
processingInformation.authorizationOptions.ignoreCvResult	Set this field to true.
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionId	
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	Set this field to 1.
processingInformation.authorizationOptions.initiator.storedCredentialUsed	Set this field to true.
processingInformation.authorizationOptions.initiator.type	Set this field to merchant.
processingInformation.authorizationOptions.transportationMode	
processingInformation.capture	Set this field to true.
processingInformation.commerceIndicator	Set this field to moto.
processingInformation.industryDataType	Set this field to transit.

Related information

- API Field Reference for the REST API

REST Example: Merchant-Initiated Sale for Debt Recovery with Stored Card Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA Debt recovery MIT sale",
    "code": "10000579",
    "transactionId": "20000579",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "commerceIndicator": "moto",
    "industryDataType": "transit",
    "reconciliationId": "1111",
    "capture": "true",
    "authorizationOptions": {
      "debtRecoveryIndicator": "true",
      "authIndicator": "1",
      "ignoreAvsResult": "true",
      "ignoreCvResult": "true",
      "transportationMode": "01",
      "initiator": {
        "type": "merchant",
        "storedCredentialUsed": "true",
        "merchantInitiatedTransaction": {
          "reason": "1",
          "previousTransactionId": "016153570198200"
        }
      }
    }
  },
  "paymentInformation": {
    "card": {
      "number": "541333XXXXXXXXXX",
      "expirationMonth": "12",
      "expirationYear": "2049",
      "type": "002"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  }
}
```

Response to a Declined Request

```
{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508879024886569004002"
    }
  },
  "clientReferenceInformation": {
    "code": "10000579",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000579"
  },
  "errorInformation": {
    "reason": "PROCESSOR_DECLINED",
    "message": "Decline - General decline of the card. No other information provided by the issuing bank."
  },
  "id": "6508879024886569004002",
  "pointOfSaleInformation": {
    "emv": {
      "tags":
 "9F360200159100AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "162971",
    "networktransactionId": "016153570198200",
    "retrievalReferenceNumber": "211511162971",
    "transactionId": "016153570198200",
        "responseCode": "05",
        "avs": {
            "code": "1"
        }
    },
    "status": "DECLINED"
}
```

Credit with a Token

This section describes how to process a stand-alone credit.

> **Important**
>
> Follow these guidelines to prevent unauthorized credits.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/credits/
Test: POST https://apitest.cybersource.com/pts/v2/credits/

Required Fields for a Credit with a Token Using the REST API

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.type

paymentInformation.instrumentIdentifier.id

Related Information

- *API field reference guide for the REST API*

REST Example: Credit with a Token

Request

```
{
    "clientReferenceInformation": {
        "code": "1234567hgh8"
    },
    "paymentInformation": {
        "instrumentIdentifier": {
            "id": "D6B858CAD38A1B7CE0531D588D0ADB5D"
        },
        "card": {
```

```
      "expirationMonth": "03",
      "expirationYear": "2031",
      "type": "062"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "225.00",
      "currency": "THB"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/credits/6538345097226017503265/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/credits/6538345097226017503265"
    }
  },
  "clientReferenceInformation": {
    "code": "1234567hgh8"
  },
  "creditAmountDetails": {
    "currency": "TBH",
    "creditAmount": "225.00"
  },
  "id": "6538345097226017503265",
  "orderInformation": {
    "amountDetails": {
      "currency": "TBH"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "062"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "062"
    },
    "instrumentIdentifier": {
      "id": "D6B858CAD38A1B7CE0531D588D0ADB5D",
      "state": "ACTIVE"
    },
    "card": {
      "type": "062"
    }
  },
  "reconciliationId": "6538345097226017503265",
  "status": "PENDING",
  "submitTimeUtc": "2024-04-25T12:06:44Z"
}
```

Capture an Authorization

This section describes how to process a capture.
When a transaction is below the threshold for First Ride Risk protection, use the capture service to capture funds from a declined authorization.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/captures`
Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures`
The `{id}` is the transaction ID returned in the authorization response.

Required Fields for a Mass Transit Capture

clientReferenceInformation.comments	
clientReferenceInformation.partner.thirdPartyCertificationNumber	**The value for this field is provided by Cybersource.**
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	

Related Information

- *API field reference guide for the REST API*

REST Example: Capturing an Authorization

Request

```
{
  "clientReferenceInformation": {
     "comments": "TransitDA BAU capture",
     "transactionId": "14987654321",
     "partner": {
        "thirdPartyCertificationNumber": "123456789012"
     }
   },
  "orderInformation": {
   "amountDetails": {
     "totalAmount": "10.00",
     "currency": "EUR"
   }
  }
}
```

Response to a Successful Request

```
{
   "_links": {
      "void": {
         "method": "POST",
         "href": "/pts/v2/captures/6484688186356910704004/voids"
      },
      "self": {
         "method": "GET",
         "href": "/pts/v2/captures/6484688186356910704004"
      }
   },
   "clientReferenceInformation": {
      "comments": "capture",
      "code": "testcode1012",
      "transactionId": "14987654321"
   },
   "id": "6484688186356910704004",
   "orderInformation": {
      "amountDetails": {
         "totalAmount": "10.00",
         "currency": "EUR"
      }
   },
   "reconciliationId": "fgssgsgsgsfg",
   "status": "PENDING",
   "submitTimeUtc": "2022-03-28T12:00:18Z"
}
```

Authorization Reversal

This section describes how to reverse an authorization.
Use the authorization reversal service to reverse an unnecessary or undesired authorization.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/reversals`
Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/reversals`
The `{id}` is the transaction ID returned in the authorization response.

Required Fields for a Mass Transit Authorization Reversal

clientReferenceInformation.comments

clientReferenceInformation.partner.thirdPartyCertificationNumber

clientReferenceInformation.transactionId

reversalInformation.amountDetails.currency

reversalInformation.amountDetails.totalAmount

Related Information

- *API field reference guide for the REST API*

REST Example: Reversing a Mass Transit Authorization

Request

```
{
  "clientReferenceInformation": {
    "comments": "REVERSAL Timeout",
    "transactionId": "11987654321",
    "partner": {
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "reversalInformation": {
    "amountDetails": {
      "totalAmount": "300.00",
      "currency": "EUR"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/reversals/6484678664766823004004"
    }
  },
  "clientReferenceInformation": {
    "code": "123456",
    "transactionId": "11987654321"
  },
  "id": "6484678664766823004004",
  "orderInformation": {
    "amountDetails": {
      "currency": "EUR"
    }
  },
  "processorInformation": {
    "responseDetails": "ABC",
    "responseCode": "00"
  },
  "reconciliationId": "6484678664766823004004",
  "reversalAmountDetails": {
    "reversedAmount": "300.00",
    "currency": "EUR"
  },
  "status": "REVERSED",
  "submitTimeUtc": "2022-03-28T11:44:26Z"
}
```

Time-Out Reversal

This section describes how to reverse an authorization that is not completed within the time allowed and times out.

Production: `POST https://api.cybersource.com/pts/v2/reversals`
Test: `POST https://apitest.cybersource.com/pts/v2/reversals`

Required Fields for a Time-Out Reversal

clientReferenceInformation.comments	**Set this field to `REVERSAL Timeout`.**
clientReferenceInformation.transactionId	
reversalInformation.amountDetails.totalAmount	
reversalInformation.reason	

Related information

- API Field Reference for the REST API

REST Example: Time-Out Reversal

Request

```
{
 "clientReferenceInformation": {
  "comments": "REVERSAL Timeout",
  "transactionId": "78885555"
 },
 "reversalInformation": {
  "amountDetails": {
   "totalAmount": "10.00"
  },
  "reason": "testing"
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "self": {
   "method": "GET",
   "href": "/pts/v2/reversals/6502854707106431104004"
  }
  },
  "clientReferenceInformation": {
    "code": "1650285470690",
    "transactionId": "78885555"
  },
  "id": "6502854707106431104004",
  "orderInformation": {
    "amountDetails": {
      "currency": "EUR"
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"5004564953419F26087C14E9BE1F1065094F07A0000000031010820220009F360203709F0702C0809F2701409F100706010A
    }
  },
  "processorInformation": {
    "responseCode": "00"
  },
  "reconciliationId": "6502854707106431104004",
  "reversalAmountDetails": {
    "reversedAmount": "10.00",
    "currency": "EUR"
  },
  "status": "REVERSED",
  "submitTimeUtc": "2022-04-18T12:37:50Z"
}
```

Response to a Decline Request

```
{
  "id": "6502857670496139204005",
  "submitTimeUtc": "2022-04-18T12:42:47Z",
  "status": "INVALID_REQUEST",
  "reason": "INVALID_DATA",
  "message": "Declined - One or more fields in the request contains invalid data"
}
```

Time-Out Void

This section describes how to void an authorization, capture, refund, or credit when you do not receive a response within the time allowed and the transaction times out. To use this feature, you must include a unique value in the **clientReferenceInformation.transactionId** field in your payment, capture, refund, or credit request and use the same unique value for the **clientReferenceInformation.transactionId** field in this request to reverse the transaction.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/voids/
Test: POST https://apitest.cybersource.com/pts/v2/voids/

Required Field for a Time-Out Void

clientReferenceInformation.comments	**Set this field to `REVERSAL Timeout`.**
clientReferenceInformation.transactionId	

Related information
- API Field Reference for the REST API

REST Example: Time-Out Void

Request

```
{
 "clientReferenceInformation": {
  "comments": "VOID Timeout",
  "transactionId": "888858556"
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "self": {
   "method": "GET",
   "href": "/pts/v2/voids/6502849034136438604002"
  }
 },
 "clientReferenceInformation": {
  "code": "1650284903396",
  "transactionId": "888858556"
 },
 "id": "6502849034136438604002",
 "orderInformation": {
  "amountDetails": {
   "currency": "EUR"
  }
 },
 "status": "VOIDED",
 "submitTimeUtc": "2022-04-18T12:28:23Z",
 "voidAmountDetails": {
  "currency": "EUR",
  "voidAmount": "10.00"
 }
}
```

Response to a Declined Request

```
{
 "id": "6502858209346457804004",
 "submitTimeUtc": "2022-04-18T12:43:41Z",
 "status": "INVALID_REQUEST",
 "reason": "INVALID_DATA",
  "message": "Declined - One or more fields in the request contains invalid data"
}
```

Debit and Prepaid Card Processing

This section shows you how to process authorizations that use a debit or prepaid card.

Requirements

In Canada, to process domestic debit transactions on Visa Platform Connect with Mastercard, you must contact customer support to have your account configured for this feature.

Related Information

- See *Debit and Prepaid Card Payments* on page 38 for a description of the debit or prepaid card transactions you can process.

Additional Resources for Debit and Prepaid Payments

For more information, see these guides:

- *PIN Debit Processing in Card Present Connect | Retail Integration Guide*
- *API field reference guide for the REST API*
- Github repositories: *https://github.com/Cybersource*

Processing Debit and Prepaid Authorizations

This section shows you how to process an authorization using debit and prepaid cards.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing Debit and Prepaid Authorizations

Use these required fields for processing debit and prepaid authorizations.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

clientReferenceInformation.code

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.type

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

Related Information

- *API field reference guide for the REST API*

Country Specific Required Fields to Process Debit and Prepaid Authorizations

Use these country-specific required fields to process a debit or prepaid authorization.

Argentina

merchantInformation.taxId	**Required for Mastercard transactions.**
merchantInformation.transactionLocalDateTime	**Required in Argentina when the time zone is not included in your account. Otherwise, this field is optional.**

Brazil

paymentInformation.card.sourceAccountType	**Required for combo card transactions.**
paymentInformation.card.sourceAccountTypeDetails	**Required for combo card line-of-credit and prepaid-card transactions.**

Chile

merchantInformation.taxId	**Required for Mastercard transactions.**

Paraguay

merchantInformation.taxId	**Required for Mastercard transactions.**

Saudi Arabia

processingInformation.authorizationOptions.transactionMode

Taiwan

paymentInformation.card.hashedNumber

Related Information

- *API field reference guide for the REST API*

Optional Field for Processing Debit and Prepaid Authorizations

You can use this optional field to include additional information when processing debit and prepaid authorizations.

processingInformation.linkId	**Set this field to the request ID that was returned in the response message from the original authorization request.**

Related Information

- *API field reference guide for the REST API*

REST Example: Processing Debit and Prepaid Authorizations

Request

```
{
  "orderInformation" : {
    "billTo" : {
      "country" : "US",
      "firstName" : "John",
      "lastName" : "Deo",
      "address1" : "901 Metro Center Blvd",
      "postalCode" : "40500",
      "locality" : "Foster City",
      "administrativeArea" : "CA",
      "email" : "test@cybs.com"
},
    "amountDetails" : {
      "totalAmount" : "100.00",
      "currency" : "USD"
    }
  },
  "paymentInformation" : {
    "card" : {
      "expirationYear" : "2031",
      "number" : "4111111111111111",
      "securityCode" : "123",
      "expirationMonth" : "12",
      "type" : "001"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links" : {
    "authReversal" : {
      "method" : "POST",
      "href" : "/pts/v2/payments/6595482584316313203494/reversals"
    },
    "self" : {
      "method" : "GET",
      "href" : "/pts/v2/payments/6595482584316313203494"
    },
    "capture" : {
      "method" : "POST",
      "href" : "/pts/v2/payments/6595482584316313203494/captures"
    }
  },
  "clientReferenceInformation" : {
    "code" : "RTS-Auth"
  },
  "consumerAuthenticationInformation" : {
    "token" : "Axj/7wSTZYq1MhJBMfMmAEQs2auWrRwyauGjNi2ZsWbJgzaOWiaVA+JbK
          AU0qB8S2VpA6cQIp4ZNvG2YbC9eM4E5NlirUyEkEx8yYAAA4A1c"
  },
 "id" : "6595482584316313203494",
 "orderInformation" : {
  "amountDetails" : {
   "authorizedAmount" : "100.00",
   "currency" : "USD"
  }
 },
 "paymentAccountInformation" : {
  "card" : {
   "type" : "001"
  }
 },
 "paymentInformation" : {
  "tokenizedCard" : {
   "type" : "001"
  },
  "card" : {
   "type" : "001"
  }
 },
 "processorInformation" : {
  "systemTraceAuditNumber" : "853428",
  "approvalCode" : "831000",
  "cardVerification" : {
   "resultCodeRaw" : "M",
   "resultCode" : "M"
  },
  "merchantAdvice" : {
   "code" : "01",
   "codeRaw" : "M001"
  },
  "responseDetails" : "ABC",
  "networkTransactionId" : "016153570198200",
  "retrievalReferenceNumber" : "221517853428",
  "consumerAuthenticationResponse" : {
   "code" : "2",
   "codeRaw" : "2"
  },
  "transactionId" : "016153570198200",
  "responseCode" : "00",
  "avs" : {
   "code" : "Y",
   "codeRaw" : "Y"
  }
 }
}
```

Enabling Debit and Prepaid Partial Authorizations

Partial authorizations and balance responses are special features that are available for debit cards and prepaid cards. This section shows you how to enable partial authorizations for a specific transaction.

To globally process domestic debit transactions on Visa Platform Connect with Mastercard in Canada, you must contact customer support to have your account configured for this feature.

Field Specific to this Use Case

Include this field in addition to the fields required for a standard authorization request:

- Indicate that this request is a partial authorization.
  Set the **processingInformation.authorizationOptions.partialAuthIndicator** to `true`.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Enabling Debit and Prepaid Partial Authorizations

Use these required fields for enabling debit and prepaid partial authorizations.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

clientReferenceInformation.code

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.type	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.partialAuthIndicator	Set the value to `true`.

Related Information

- *API field reference guide for the REST API*

Optional Field for Enabling Debit and Prepaid Partial Authorizations

You can use these optional fields to include additional information when enabling debit and prepaid partial authorizations.

processingInformation.linkId	**Set this field to the request ID that was returned in the response message from the original authorization request.**

Related Information

- *API field reference guide for the REST API*

REST Example: Enabling Debit and Prepaid Partial Authorizations

Request

```
{
  "clientReferenceInformation" : {
    "code" : "TC50171_3"
  },
  "orderInformation" : {
    "billTo" : {
      "country" : "US",
      "lastName" : "Deo",
      "address2" : "Address 2",
      "address1" : "201 S. Division St.",
      "postalCode" : "48104-2201",
      "locality" : "Ann Arbor",
      "administrativeArea" : "MI",
      "firstName" : "John",
      "phoneNumber" : "999999999",
      "district" : "MI",
      "buildingNumber" : "123",
      "company" : "Visa",
   "email" : "test@cybs.com"
  },
  "amountDetails" : {
   "totalAmount" : "1000.00",
   "currency" : "USD"
  }
 },
 "paymentInformation" : {
  "card" : {
   "expirationYear" : "2031",
   "number" : "5555555555xxxxxx",
   "securityCode" : "123",
   "expirationMonth" : "12",
   "type" : "002"
  }
 },
"processingInformation" : {
 "authorizationOptions" : {
   "partialAuthIndicator" : "true"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links" : {
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6595549144566655003494"
  }
 },
 "clientReferenceInformation" : {
  "code" : "TC50171_3"
 },
 "id" : "6595549144566655003494",
 "orderInformation" : {
  "amountDetails" : {
   "totalAmount" : "1000.00",
   "authorizedAmount" : "499.01",
   "currency" : "USD"
  }
 },
 "paymentInformation" : {
  "accountFeatures" : {
   "currency" : "usd",
   "balanceAmount" : "0.00"
  }
 },
 "pointOfSaleInformation" : {
  "terminalId" : "261996"
 },
 "processorInformation" : {
  "merchantNumber" : "000000092345678",
  "approvalCode" : "888888",
  "cardVerification" : {
   "resultCode" : ""
  },
  "networkTransactionId" : "123456789619999",
  "transactionId" : "123456789619999",
  "responseCode" : "100",
  "avs" : {
   "code" : "X",
   "codeRaw" : "I1"
  }
 },
 "reconciliationId" : "56059417N6C86KTJ",
 "status" : "PARTIAL_AUTHORIZED",
 "submitTimeUtc" : "2022-08-03T19:28:34Z"
}
```

Disabling Debit and Prepaid Partial Authorizations

This topic shows you how to successfully disable partial authorizations for specific transactions.

Field Specific to this Use Case

Include this field in addition to the fields required for a standard authorization request:

- Indicate that this request is not a partial authorization.
  Set the `processingInformation.authorizationOptions.partialAuthIndicator` to `false`.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Field for Disabling Debit and Prepaid Partial Authorizations

Use these required fields for disabling debit and prepaid partial authorizations.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

clientReferenceInformation.code

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.type

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

processingInformation.authorizationOptions.partialAuthIndic

Set the value to `false` in an authorization or sale request. When you do so, only that specific transaction is disabled for partial authorization.

Related Information

- *API field reference guide for the REST API*

Optional Field for Disabling Debit and Prepaid Partial Authorizations

You can use this optional field to include additional information when disabling debit and prepaid partial authorizations.

processingInformation.linkId

Set this field to the request ID that was returned in the response message from the original authorization request.

Related Information

- *API field reference guide for the REST API*

REST Example: Disabling Debit and Prepaid Partial Authorizations

Request

```
{
   "processingInformation":{
  "authorizationOptions":{
   "partialAuthIndicator": "false"
  }
 },
  "clientReferenceInformation" : {
    "code" : "TC50171_3"
  },
  "orderInformation" : {
    "billTo" : {
      "country" : "US",
      "lastName" : "Deo",
      "address2" : "Address 2",
      "address1" : "201 S. Division St.",
      "postalCode" : "48104-2201",
      "locality" : "Ann Arbor",
      "administrativeArea" : "MI",
      "firstName" : "John",
      "phoneNumber" : "999999999",
      "district" : "MI",
      "buildingNumber" : "123",
      "company" : "Visa",
      "email" : "test@cybs.com"
    },
    "amountDetails" : {
      "totalAmount" : "501.00",
      "currency" : "USD"
    }
  },
  "paymentInformation" : {
    "card" : {
      "expirationYear" : "2031",
      "number" : "5555555555xxxxxx",
      "securityCode" : "123",
      "expirationMonth" : "12",
      "type" : "002"
    }
  }
}
```

Response to a Successful Request

```
{
    "_links": {
        "self": {
            "method": "GET",
            "href": "/pts/v2/payments/6595545423896900104953"
        }
    },
    "clientReferenceInformation": {
        "code": "TC50171_3"
    },
    "errorInformation": {
        "reason": "PROCESSOR_DECLINED",
        "message": "Decline - General decline of the card.
                    No other information provided by the issuing bank."
    },
    "id": "6595545423896900104953",
    "pointOfSaleInformation": {
```

```
    "terminalId": "111111"
  },
  "processorInformation": {
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "status": "DECLINED"
}
```

Airline Data Processing

This section describes how to process airline payments.

Requirement

When you are ready to go live with airline data processing, contact Cybersource Customer Support to have your account configured to process airline data. If your account is not enabled, and you try to send airline transactions, you will receive an error for invalid data.

Related Information

- See *Airline Data* on page 39 for information about and requirements for processing payments that include airline data.
- See *Airline Data Reference Information* on page 41 for a list and description of the different document type and ancillary service category codes that are used when processing payments that include airline data.

Additional Resources for Airline Data

For more information, see these guides:

- *Airline Processing Developer Guide*
- *API field reference guide for the REST API*
- Github repositories: *https://github.com/Cybersource*

Airline Travel Legs

Some processors require travel legs in the API service request, even for direct flights. This section describes how to successfully include travel legs in an API request.

Using Travel Legs

To include travel legs in an airline transaction, include one or more travel legs in the **legs[]** array.

For example, these three travel legs are valid:

```
"travelInformation": {
  "transit": {
   "airline": {
    "legs": [
     {
      "carrierCode": "XX"
     },
     {
      "carrierCode": "XZ"
     },
     {
      "carrierCode": "XX"
     }
    ]
   }
```

Travel Leg Limitations

Some processors limit the number of travel legs for each trip based on the card type.

Authorize an Airline Ticket Payment

This section describes how to process an airline authorization.

Authorization Restrictions

Ticket purchases that include multiple passengers may be included in a single authorization request, but you must make separate capture requests for every passenger.

If any ancillary purchases are made at the same time as the ticket purchase, you may include all items in a single authorization request, but you must separate the ancillary and ticket purchases into their own capture requests.

If any ancillary purchases are made not at the same time as the ticket purchase, you must send separate authorization and capture requests for the ancillary and ticket purchases.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`

Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing an Airline Payment

Include these required fields for authorizing an airline payment.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.industryDataType	Set to `airline`.

REST Example: Authorizing an Airline Payment

Request

```
{
  "processingInformation": {
    "industryDataType": "airline"
  },
  "paymentInformation": {
    "card": {
      "number": "4111111111111111",
      "expirationMonth": "12",
      "expirationYear": "2031"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "500.00",
      "currency": "usd"
    },
    "billTo": {
```

```
   "firstName": "John",
   "lastName": "Doe",
   "address1": "123 Happy St.",
   "locality": "Sunny Town",
   "administrativeArea": "CA",
   "postalCode": "12345-1234",
   "country": "US",
   "email": "test@cybs.com"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6823009451126309503954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6823009451126309503954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6823009451126309503954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1682300945230"
 },
 "id": "6823009451126309503954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "500.00",
   "currency": "usd"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "67720603YGMSE5JE",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-04-24T01:49:05Z"
}
```

Captures for Ticket Purchases

This section describes how to capture an airline payment for ticket purchases. Captures for ticket purchases must be made separately from captures for ancillary purchases. For more information about how to capture an ancillary purchase, see *Captures for Ancillary Purchases* on page 221.

Travel Legs

You can use travel leg fields for trips that have multiple legs. For more information on how to use travel leg fields, see *Airline Data* on page 39.

Leg Limitations

Visa Platform Connect limits the maximum number of legs for each trip based on card type. This table describes the maximum number of legs for each trip based on card type.

Visa Platform Connect Leg Limitations

Supported Card Types	Maximum Number of Trip Legs
American Express	4
Discover	4
Mastercard	4
Visa	4

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/captures`

Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures`

The `{id}` is the transaction ID returned in the authorization response.

Required Fields for Capturing an Airline Payment

Include these required fields to capture an airline payment for ticket purchases.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
processingInformation.industryDataType	**Set to `airline`.**

Card-Specific Field to Capture an Airline Payment

This section includes card-specific information.

Mastercard

Use this card-specific field in addition to the required fields when capturing an authorization with a Mastercard.

travelInformation.transit.airline.ticketIssuer.code	

Optional Fields for Capturing an Airline Payment

You can use these optional fields to include additional information when capturing an airline payment.

orderInformation.amountDetails.taxAmount	
orderInformation.amountDetails.taxDetails[].amount	
orderInformation.lineItems[].taxDetails[].code	
orderInformation.lineItems[].totalAmount	
travelInformation.agency.code	
travelInformation.agency.name	
travelInformation.transit.airline.customerCode	
travelInformation.transit.airline.documentType	**For a list of possible values, see *Airline Document Type Codes* on page 41.**
travelInformation.transit.airline.exchangeTicketFeeAmount	
travelInformation.transit.airline.legs[].conjunctionTicket	
travelInformation.transit.airline.legs[].couponNumber	
travelInformation.transit.airline.legs[].endorsementsRestrictions	
travelInformation.transit.airline.legs[].exchangeTicketNumber	
travelInformation.transit.airline.legs[].fareBasis	
travelInformation.transit.airline.legs[].feeAmount	
travelInformation.transit.airline.legs[].stopoverIndicator	

travelInformation.transit.airline.legs[].taxAmount

travelInformation.transit.airline.planNumber

travelInformation.transit.airline.ticketChangeIndicator

travelInformation.transit.airline.ticketIssueDate

travelInformation.transit.airline.totalFeeAmount

travelInformation.transit[].exchangeTicketAmount

Related Information

- *API field reference guide for the REST API*

Example: Capturing an Airline Payment

Use this example as a reference for capturing an airline payment.

Request

```
{
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "processingInformation": {
    "industryDataType": "airline"
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "500.00",
      "currency": "USD"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/captures/6823025890736075903954/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/captures/6823025890736075903954"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "id": "6823025890736075903954",
  "orderInformation": {
    "amountDetails": {
```

```
   "totalAmount": "500.00",
   "currency": "USD"
  }
 },
 "reconciliationId": "67720603YGMSE5JE",
 "status": "PENDING",
 "submitTimeUtc": "2023-04-24T02:16:29Z"
}
```

Captures for Ancillary Purchases

This section describes how to capture an airline payment for ancillary purchases. Ancillary purchases are any additional services, such as baggage, meals, and paid seats, that your customers can purchase. Captures for ancillary purchases must be made separately from captures for ticket purchases.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/captures`
Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures`
The `{id}` is the transaction ID returned in the authorization response.

Required Fields for Capturing an Authorization for Ancillary Purchases

Include these required fields to capture an airline payment for ancillary purchases.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
processingInformation.industryDataType	**Set to `airline`.**

Related Information

- *API field reference guide for the REST API*

Ancillary Fields for Capturing an Authorization for an Ancillary Purchase

Choose from these optional ancillary fields to add additional information when capturing an ancillary purchase.

travelInformation.transit.airline.ancillaryInformation.connectedTicketNumber

travelInformation.transit.airline.ancillaryInformation.creditReasonIndicator

travelInformation.transit.airline.ancillaryInformation.passengerName

travelInformation.transit.airline.ancillaryInformation.service[].categoryCode

For a list of possible values, see *Ancillary Service Category Codes* **on page 44.**

travelInformation.transit.airline.ancillaryInformation.service[].subCategoryCode

travelInformation.transit.airline.ancillaryInformation.ticketNumber

REST Example: Capturing an Authorization for an Ancillary Purchase

Use this example as a reference for capturing an ancillary purchase with the ancillary fields.

Request

```
{
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "processingInformation": {
  "industryDataType": "airline"
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "500.00",
   "currency": "USD"
  }
 },
 "travelInformation": {
  "transit": {
   "airline": {
    "ancillaryInformation": {
     "ticketNumber": "123456789123456",
     "passengerName": "John Doe",
     "connectedTicketNumber": "654321987654321"
    }
   }
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
   "href": "/pts/v2/captures/6823030661646093703954/voids"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/captures/6823030661646093703954"
  }
 },
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
  "id": "6823030661646093703954",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "500.00",
      "currency": "USD"
    }
  },
  "reconciliationId": "67221841NGMV8WOT",
  "status": "PENDING",
  "submitTimeUtc": "2023-04-24T02:24:26Z"
}
```

Refunds

This topic describes how to process an airline refund.
This service returns the funds used in the initial capture and requires the original capture ID.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/refunds`
Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/refunds`
The `{id}` is the transaction ID returned in the capture or sale response.

Required Fields for Processing an Airline Refund

Include these required fields to process an airline refund.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

Related Information

- *API field reference guide for the REST API*

Optional Fields for Processing an Airline Refund

This section includes these types of optional fields for an airline refund:

- *General Optional Fields*
- *Optional Fields for Ticket Purchases*
- *Optional Fields for Ancillary Purchases*

General Optional Fields

You can use these optional fields to include additional information in any airline purchase.

travelInformation.agency.code

travelInformation.agency.name

travelInformation.transit.airline.arrivalDate

travelInformation.transit.airline.carrierName

travelInformation.transit.airline.clearingCount

travelInformation.transit.airline.clearingSequence

travelInformation.transit.airline.creditReasonIndicator

travelInformation.transit.airline.customerCode

travelInformation.transit.airline.documentType For a list of possible values, see *Airline Document Type Codes* on page 41.

travelInformation.transit.airline.electronicTicketIndicator

travelInformation.transit.airline.exchangeTicketFeeAmount

travelInformation.transit.airline.numberOfPassengers

travelInformation.transit.airline.passengerName

travelInformation.transit.airline.planNumber

travelInformation.transit.airline.purchaseType

travelInformation.transit.airline.reservationSystemCode

travelInformation.transit.airline.restrictedTicketDesciption

travelInformation.transit.airline.restrictedTicketIndicator

travelInformation.transit.airline.ticketChangeIndicator

travelInformation.transit.airline.ticketIssueDate

travelInformation.transit.airline.ticketIssuer.locality

travelInformation.transit.airline.ticketNumber

travelInformation.transit.airline.totalClearingAmount

travelInformation.transit.airline.totalFeeAmount

travelInformation.transit[].exchangeTicketAmount

Airline Optional Fields for Ticket Purchases

You can use these optional fields to include additional information when requesting an airline credit for a ticket purchase.

travelInformation.transit.airline.legs[].arrivalTime

travelInformation.transit.airline.legs[].arrivalTimeMeridian

travelInformation.transit.airline.legs[].carrierCode

travelInformation.transit.airline.legs[].class

travelInformation.transit.airline.legs[].conjunctionTicket

travelInformation.transit.airline.legs[].couponNumber

travelInformation.transit.airline.legs[].departureDate

travelInformation.transit.airline.legs[].departureTime

travelInformation.transit.airline.legs[].departureTimeMeridian

travelInformation.transit.airline.legs[].destinationAirportCode

travelInformation.transit.airline.legs[].endorsementsRestrictions

travelInformation.transit.airline.legs[].exchangeTicketNumber

travelInformation.transit.airline.legs[].fareBasis

travelInformation.transit.airline.legs[].feeAmount

travelInformation.transit.airline.legs[].flightNumber

travelInformation.transit.airline.legs[].originatingAirportCode

travelInformation.transit.airline.legs[].stopoverIndicator

travelInformation.transit.airline.legs[].taxAmount

travelInformation.transit.airline.legs[].totalFareAmount

Ancillary Optional Fields

You can use these optional fields to include additional information when requesting an airline credit for an ancillary purchase.

travelInformation.transit.airline.ancillaryInformation.connectedTicketNumber

travelInformation.transit.airline.ancillaryInformation.creditReasonIndicator

travelInformation.transit.airline.ancillaryInformation.passengerName

travelInformation.transit.airline.ancillaryInformation.service[].categoryCode

For a list of possible values, see *Ancillary Service Category Codes* **on page 44.**

travelInformation.transit.airline.ancillaryInformation.service[].subCategoryCode

travelInformation.transit.airline.ancillaryInformation.ticketNumber

Related Information

- *API field reference guide for the REST API*

REST Example: Processing an Airline Refund

Use this example as a reference for processing an airline refund.

Request

```
{
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "processingInformation": {
    "industryDataType": "airline"
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "500",
      "currency": "USD"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/refunds/6823038625416445403955/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/refunds/6823038625416445403955"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "id": "6823038625416445403955",
  "orderInformation": {
    "amountDetails": {
      "currency": "USD"
    }
  },
  "processorInformation": {
    "approvalCode": "888888",
    "responseCode": "100"
  },
  "reconciliationId": "67722608EGMV6Q7V",
  "refundAmountDetails": {
    "currency": "USD",
    "refundAmount": "500.00"
  },
  "status": "PENDING",
  "submitTimeUtc": "2023-04-24T02:37:42Z"
}
```

Issue a Refund

This topic describes how to process an airline credit.
This service distributes funds without requiring a capture ID.

Important

All fields used in the original transaction must be included in your request.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/credits/`
Test: `POST https://apitest.cybersource.com/pts/v2/credits/`

Required Fields for Processing an Airline Credit

Include these required fields to process an airline credit.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

Optional Fields for Processing an Airline Credit

This section includes these types of optional fields for an airline credit:

- *General Optional Fields*
- *Optional Fields for Ticket Purchases*
- *Optional Fields for Ancillary Purchases*

General Optional Fields

You can use these optional fields to include additional information in any airline purchase.

travelInformation.agency.code

travelInformation.agency.name

travelInformation.transit.airline.arrivalDate

travelInformation.transit.airline.carrierName

travelInformation.transit.airline.clearingCount

travelInformation.transit.airline.clearingSequence

travelInformation.transit.airline.creditReasonIndicator

travelInformation.transit.airline.customerCode

travelInformation.transit.airline.documentType

For a list of possible values, see *Airline Document Type Codes* on page 41.

travelInformation.transit.airline.electronicTicketIndicator

travelInformation.transit.airline.exchangeTicketFeeAmount

travelInformation.transit.airline.numberOfPassengers

travelInformation.transit.airline.passengerName

travelInformation.transit.airline.planNumber

travelInformation.transit.airline.purchaseType

travelInformation.transit.airline.reservationSystemCode

travelInformation.transit.airline.restrictedTicketDesciption

travelInformation.transit.airline.restrictedTicketIndicator

travelInformation.transit.airline.ticketChangeIndicator

travelInformation.transit.airline.ticketIssueDate

travelInformation.transit.airline.ticketIssuer.locality

travelInformation.transit.airline.ticketNumber

travelInformation.transit.airline.totalClearingAmount

travelInformation.transit.airline.totalFeeAmount

travelInformation.transit[].exchangeTicketAmount

Airline Optional Fields for Ticket Purchases

You can use these optional fields to include additional information when requesting an airline credit for a ticket purchase.

travelInformation.transit.airline.legs[].arrivalTime

travelInformation.transit.airline.legs[].arrivalTimeMeridian

travelInformation.transit.airline.legs[].carrierCode

travelInformation.transit.airline.legs[].class

travelInformation.transit.airline.legs[].conjunctionTicket

travelInformation.transit.airline.legs[].couponNumber

travelInformation.transit.airline.legs[].departureDate

travelInformation.transit.airline.legs[].departureTime

travelInformation.transit.airline.legs[].departureTimeMeridian

travelInformation.transit.airline.legs[].destinationAirportCode

travelInformation.transit.airline.legs[].endorsementsRestrictions

travelInformation.transit.airline.legs[].exchangeTicketNumber

travelInformation.transit.airline.legs[].fareBasis

travelInformation.transit.airline.legs[].feeAmount

travelInformation.transit.airline.legs[].flightNumber

travelInformation.transit.airline.legs[].originatingAirportCode

travelInformation.transit.airline.legs[].stopoverIndicator

travelInformation.transit.airline.legs[].taxAmount

travelInformation.transit.airline.legs[].totalFareAmount

Ancillary Optional Fields

You can use these optional fields to include additional information when requesting an airline credit for an ancillary purchase.

travelInformation.transit.airline.ancillaryInformation.connectedTicketNumber

travelInformation.transit.airline.ancillaryInformation.creditReasonIndicator

travelInformation.transit.airline.ancillaryInformation.passengerName

travelInformation.transit.airline.ancillaryInformation.service[].categoryCode

For a list of possible values, see *Ancillary Service Category Codes* **on page 44.**

travelInformation.transit.airline.ancillaryInformation.service[].subCategoryCode

travelInformation.transit.airline.ancillaryInformation.ticketNumber

REST Example: Processing an Airline Credit

Request

```
{
 "paymentInformation": {
  "card": {
   "number": "4111111111111111",
   "expirationMonth": "12",
   "expirationYear": "31"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "500.00",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "123 Happy St.",
   "locality": "Sunnyville",
   "administrativeArea": "CA",
   "postalCode": "12345",
   "country": "US",
   "email": "johndoe@test.com"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
   "href": "/pts/v2/credits/6823065885666134104951/voids"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/credits/6823065885666134104951"
  }
 },
 "clientReferenceInformation": {
  "code": "1682306588644"
 },
 "creditAmountDetails": {
  "currency": "USD",
  "creditAmount": "500.00"
 },
 "id": "6823065885666134104951",
 "orderInformation": {
  "amountDetails": {
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "processorInformation": {
  "approvalCode": "888888",
  "responseCode": "100"
 },
 "reconciliationId": "74259417PGM9TXHT",
 "status": "PENDING",
 "submitTimeUtc": "2023-04-24T03:23:08Z"
}
```

Japanese Payment Options Processing

This section shows you how to process an authorization with Japanese payment options (JPO).
JPO supports these payment methods:

- Single payment
- Bonus payment
- Installment payment
- Revolving payment
- Combination of bonus payment and installment payment

Requirements

- You have signed a contract with your acquirer.
- You have contacted your account provider for details about contracts and funding cycles. The funding cycle could differ when using JPO.
- Card holders who want to use JPO have signed a contract with an issuing bank.
- You have confirmed payment option availability with your account provider and card holder before implementing one of these payment options.

Related Information

- See *Japanese Payment Options* on page 46 for a description of JPO payments.

Authorize a Single Payment with Japanese Payment Options

This section shows you how to process an authorization of a single payment with Japanese Payment Options (JPO).

Limitations

- The only supported acquirer is Sumitomo Mitsui Card Co.
- The payment must use a Visa payment card issued in Japan, and the only supported acquirer is Sumitomo Mitsui Card Co.

Prerequisites

- You have signed a contract with your acquirer.
- You have contacted your account provider for details about contracts and funding cycles. The funding cycle could differ when using JPO.
- Card holders who want to use JPO have signed a contract with an issuing bank.
- You have confirmed payment option availability with your account provider and card holder before implementing one of these payment options.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Single Payment Using the JPO Method

Use these required fields for authorizing a single payment using the JPO method.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
paymentInformation.card.type	
processingInformation.japanPaymentOptions.businessName	**Business name in kanji characters.**
processingInformation.japanPaymentOptions.businessNameAlphaNumeric	
processingInformation.japanPaymentOptions.businessNameKatakana	
processingInformation.japanPaymentOptions.terminalId	**Required for card-present transactions. Unique Japan Credit Card Association (JCCA) terminal identifier that is provided by Cybersource.**

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a JPO Single Payment

Request

```
{
   "orderInformation": {
      "billTo": {
         "country": "US",
         "lastName": "Kim",
         "address1": "201 S. Division St.",
         "postalCode": "48104-2201",
         "locality": "Ann Arbor",
         "administrativeArea": "MI",
         "firstName": "Kyong-Jin",
         "email": "test@cybs.com"
      },
      "amountDetails": {
         "totalAmount": "100.00",
         "currency": "jpy"
      }
   },
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "number": "4111111111111111",
         "expirationMonth": "12",
         "type": "001"
      }
   },
   "processingInformation": {
      "japanPaymentOptions": {
         "businessName": "##",
         "businessNameAlphaNumeric": "OurStore",
      "businessNameKatakana": "######"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links" : {
  "authReversal" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6842924689096191303059/reversals"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6842924689096191303059"
  },
  "capture" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6842924689096191303059/captures"
  }
 },
 "clientReferenceInformation" : {
  "code" : "RTS-Auth"
 },
 "id" : "6842924689096191303059",
 "orderInformation" : {
  "invoiceDetails" : {
   "salesSlipNumber" : "52966"
  },
  "amountDetails" : {
   "authorizedAmount" : "100",
   "currency" : "jpy"
  }
 },
 "paymentAccountInformation" : {
  "card" : {
   "type" : "001"
  }
 },
 "paymentInformation" : {
  "tokenizedCard" : {
   "type" : "001"
  },
  "card" : {
   "type" : "001"
  }
 },
 "processorInformation" : {
  "salesSlipNumber" : "52966",
  "approvalCode" : "123456",
  "cardVerification" : {
   "resultCode" : "3"
  },
  "responseCategoryCode" : "000",
  "forwardedAcquirerCode" : "Sumitomo",
    "avs" : {
      "code" : "2"
    }
  },
  "reconciliationId" : "00202305171201090000000000001",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2023-05-17T03:01:09Z"
}
```

Authorize a Bonus Payment with Japanese Payment Options

This section shows you how to process an authorization of a bonus payment with Japanese Payment Options (JPO).

Limitations

- The only supported acquirer is Sumitomo Mitsui Card Co.
- The payment must use a Visa payment card.

Prerequisites

- You have signed a contract with your acquirer.
- You have contacted your account provider for details about contracts and funding cycles. The funding cycle could differ when using JPO.
- Card holders who want to use JPO have signed a contract with an issuing bank.
- You have confirmed payment option availability with your account provider and card holder before implementing one of these payment options.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a JPO Bonus Payment

Use these required fields for authorizing a JPO bonus payment.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
paymentInformation.card.type	
processingInformation.japanPaymentOptions.businessName	**Business name in kanji characters.**
processingInformation.japanPaymentOptions.businessNameAlphaNumeric	
processingInformation.japanPaymentOptions.businessNameKatakana	
processingInformation.japanPaymentOptions.paymentMethod	**Set this field to** 21**,** 22**,** 23**, or** 24**.**
processingInformation.japanPaymentOptions.terminalId	**Required for card-present transactions. Unique Japan Credit Card Association (JCCA) terminal identifier that is provided by Cybersource.**

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a JPO Bonus Payment

Request

```
{
  "orderInformation": {
    "billTo": {
      "country": "US",
      "lastName": "Kim",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "firstName": "Kyong-Jin",
            "email": "test@cybs.com"
        },
        "amountDetails": {
            "totalAmount": "100.00",
            "currency": "jpy"
        }
    },
    "paymentInformation": {
        "card": {
            "expirationYear": "2031",
            "number": "4111111111111111",
            "expirationMonth": "12",
            "type": "001"
        }
    },
    "processingInformation": {
        "japanPaymentOptions": {
            "businessName": "##",
            "businessNameAlphaNumeric": "OurStore",
            "businessNameKatakana": "######",
            "paymentMethod": "21"
        }
    }
}
```

Response to a Successful Request

```
{
  "_links" : {
    "authReversal" : {
      "method" : "POST",
      "href" : "/pts/v2/payments/6843556498736135003059/reversals"
    },
    "self" : {
      "method" : "GET",
      "href" : "/pts/v2/payments/6843556498736135003059"
    },
    "capture" : {
      "method" : "POST",
      "href" : "/pts/v2/payments/6843556498736135003059/captures"
    }
  },
  "clientReferenceInformation" : {
    "code" : "RTS-Auth"
  },
  "id" : "6843556498736135003059",
  "orderInformation" : {
    "invoiceDetails" : {
      "salesSlipNumber" : "56307"
    },
    "amountDetails" : {
      "authorizedAmount" : "100",
      "currency" : "jpy"
    }
  },
  "paymentAccountInformation" : {
    "card" : {
      "type" : "001"
    }
  },
  "paymentInformation" : {
    "tokenizedCard" : {
      "type" : "001"
    },
    "card" : {
      "type" : "001"
    }
  },
  "processorInformation" : {
    "salesSlipNumber" : "56307",
    "approvalCode" : "123456",
    "cardVerification" : {
      "resultCode" : "3"
    },
    "responseCategoryCode" : "000",
    "forwardedAcquirerCode" : "Sumitomo",
    "avs" : {
      "code" : "2"
    }
  },
  "reconciliationId" : "0020230518053410000000000001",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2023-05-17T20:34:10Z"
}
```

Authorize an Installment Payment with Japanese Payment Options

This section shows you how to process an authorization of an installment payment with Japanese Payment Options (JPO).

Limitations

- The only supported acquirer is Sumitomo Mitsui Card Co.
- The payment must use a Visa payment card.

Prerequisites

- You have signed a contract with your acquirer.
- You have contacted your account provider for details about contracts and funding cycles. The funding cycle could differ when using JPO.
- Card holders who want to use JPO have signed a contract with an issuing bank.
- You have confirmed payment option availability with your account provider and card holder before implementing one of these payment options.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a JPO Installment Payment

Use these required fields for authorizing a JPO installment payment.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

paymentInformation.card.type

processingInformation.japanPaymentOptions.businessName **Business name in kanji characters.**

processingInformation.japanPaymentOptions.businessNameAlphaNumeric

processingInformation.japanPaymentOptions.businessNameKatakana

processingInformation.japanPaymentOptions.firstBillingMonth **If you do not specify this field, it is set by default to the number of the next month.**

processingInformation.japanPaymentOptions.installments **Number of monthly payments.**

processingInformation.japanPaymentOptions.paymentMethod **Set the value to `61`.**

processingInformation.japanPaymentOptions.terminalId	**Required for card-present transactions. Unique Japan Credit Card Association (JCCA) terminal identifier that is provided by Cybersource.**

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a JPO Installment Payment

Request

```
{
   "orderInformation": {
      "billTo": {
         "country": "US",
         "lastName": "Kim",
         "address1": "201 S. Division St.",
         "postalCode": "48104-2201",
         "locality": "Ann Arbor",
         "administrativeArea": "MI",
         "firstName": "Kyong-Jin",
         "email": "test@cybs.com"
      },
      "amountDetails": {
         "totalAmount": "100.00",
         "currency": "jpy"
      }
   },
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "number": "4111111111111111",
         "expirationMonth": "12",
         "type": "001"
      }
   },
   "processingInformation": {
      "japanPaymentOptions": {
         "businessName": "##",
         "businessNameAlphaNumeric": "OurStore",
         "businessNameKatakana": "######",
         "firstBusinessMonth": "04",
         "installments": "12",
         "paymentMethod": "31"
      }
   }
}
```

Response to a Successful Request

```
{
 "_links" : {
  "authReversal" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6843585327946622203059/reversals"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6843585327946622203059"
  },
  "capture" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6843585327946622203059/captures"
  }
 },
 "clientReferenceInformation" : {
  "code" : "RTS-Auth"
 },
 "id" : "6843585327946622203059",
 "orderInformation" : {
  "invoiceDetails" : {
   "salesSlipNumber" : "56311"
  },
  "amountDetails" : {
   "authorizedAmount" : "100",
   "currency" : "jpy"
  }
 },
 "paymentAccountInformation" : {
  "card" : {
   "type" : "001"
  }
 },
 "paymentInformation" : {
  "tokenizedCard" : {
   "type" : "001"
  },
  "card" : {
   "type" : "001"
  }
 },
 "processorInformation" : {
  "salesSlipNumber" : "56311",
  "approvalCode" : "123456",
  "cardVerification" : {
   "resultCode" : "3"
  },
  "responseCategoryCode" : "000",
  "forwardedAcquirerCode" : "Sumitomo",
  "avs" : {
   "code" : "2"
  }
 },
 "reconciliationId" : "00202305180622130000000000001",
 "status" : "AUTHORIZED",
 "submitTimeUtc" : "2023-05-17T21:22:13Z"
}
```

Authorize a Revolving Payment with Japanese Payment Options

This section shows you how to process an authorization of a revolving payment with Japanese Payment Options (JPO).

Limitations

- The only supported acquirer is Sumitomo Mitsui Card Co.
- The payment must use a Visa payment card.

Prerequisites

- You have signed a contract with your acquirer.
- You have contacted your account provider for details about contracts and funding cycles. The funding cycle could differ when using JPO.
- Card holders who want to use JPO have signed a contract with an issuing bank.
- You have confirmed payment option availability with your account provider and card holder before implementing one of these payment options.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Revolving Payment Using the JPO Method

Use these required fields for authorizing a revolving payment using the JPO method.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
paymentInformation.card.type	
processingInformation.japanPaymentOptions.businessName	**Business name in kanji characters.**
processingInformation.japanPaymentOptions.businessNameAlphaNumeric	
processingInformation.japanPaymentOptions.businessNameKatakana	
processingInformation.japanPaymentOptions.firstBillingMonth	**Number of the month in which installment payments begin. The default value is the number of the month that follows the transaction date.**
processingInformation.japanPaymentOptions.installments	**Set this field to the number of installment payments.**
processingInformation.japanPaymentOptions.paymentMethod	**Set the value to `80`.**
processingInformation.japanPaymentOptions.terminalId	**Required for card-present transactions. Unique Japan Credit Card Association (JCCA) terminal identifier that is provided by Cybersource.**

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a JPO Revolving Payment

Request

```
{
  "orderInformation": {
    "billTo": {
      "country": "US",
      "lastName": "Kim",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "firstName": "Kyong-Jin",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "jpy"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111111111111111",
      "expirationMonth": "12",
      "type": "001"
    }
  },
  "processingInformation": {
    "japanPaymentOptions": {
      "businessName": "##",
      "businessNameAlphaNumeric": "OurStore",
      "businessNameKatakana": "######",
      "firstBusinessMonth": "05",
      "installments": "12",
      "paymentMethod": "80"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links" : {
  "authReversal" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6843585327946622203059/reversals"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6843585327946622203059"
  },
  "capture" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6843585327946622203059/captures"
  }
 },
 "clientReferenceInformation" : {
  "code" : "RTS-Auth"
 },
 "id" : "6843585327946622203059",
 "orderInformation" : {
  "invoiceDetails" : {
   "salesSlipNumber" : "56311"
  },
  "amountDetails" : {
   "authorizedAmount" : "100",
   "currency" : "jpy"
```

```
    }
  },
  "paymentAccountInformation" : {
    "card" : {
      "type" : "001"
    }
  },
  "paymentInformation" : {
    "tokenizedCard" : {
      "type" : "001"
    },
    "card" : {
      "type" : "001"
    }
  },
  "processorInformation" : {
    "salesSlipNumber" : "56311",
    "approvalCode" : "123456",
    "cardVerification" : {
      "resultCode" : "3"
    },
    "responseCategoryCode" : "000",
    "forwardedAcquirerCode" : "Sumitomo",
    "avs" : {
      "code" : "2"
    }
  },
  "reconciliationId" : "00202305180622130000000000001",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2023-05-17T21:22:13Z"
}
```

Authorize a Combination Payment with Japanese Payment Options

This section shows you how to process an authorization of a combination bonus and installment payments with Japanese Payment Options (JPO).

Limitations

- The only supported acquirer is Sumitomo Mitsui Card Co.
- The payment must use a Visa payment card.

Prerequisites

- You have signed a contract with your acquirer.
- You have contacted your account provider for details about contracts and funding cycles. The funding cycle could differ when using JPO.
- Card holders who want to use JPO have signed a contract with an issuing bank.

- You have confirmed payment option availability with your account provider and card holder before implementing one of these payment options.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Combination Payment Using the JPO Method

Use these required fields for authorizing a combination payment using the JPO method.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

paymentInformation.card.type

processingInformation.japanPaymentOptions.businessName **Business name in kanji characters.**

processingInformation.japanPaymentOptions.businessNameAlphaNumeric

processingInformation.japanPaymentOptions.businessNameKatakana

processingInformation.japanPaymentOptions.firstBillingMonth

processingInformation.japanPaymentOptions.installments	**Set this field to the number of monthly installments.**
processingInformation.japanPaymentOptions.paymentMethod	**Set this field to 31, 32, 33, or 34.**
processingInformation.japanPaymentOptions.terminalId	**Required for card-present transactions. Unique Japan Credit Card Association (JCCA) terminal identifier that is provided by Cybersource.**

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a JPO Combination Payment

Request

```
{
  "orderInformation": {
    "billTo": {
      "country": "US",
      "lastName": "Kim",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "firstName": "Kyong-Jin",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "jpy"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111111111111111",
      "expirationMonth": "12",
      "type": "001"
    }
  },
  "processingInformation": {
    "japanPaymentOptions": {
      "businessName": "##",
      "businessNameAlphaNumeric": "OurStore",
      "businessNameKatakana": "######",
      "firstBusinessMonth": "05",
      "installments": "12",
      "paymentMethod": "80"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links" : {
  "authReversal" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6843585327946622203059/reversals"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6843585327946622203059"
  },
  "capture" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6843585327946622203059/captures"
  }
 },
 "clientReferenceInformation" : {
  "code" : "RTS-Auth"
 },
 "id" : "6843585327946622203059",
 "orderInformation" : {
  "invoiceDetails" : {
   "salesSlipNumber" : "56311"
  },
  "amountDetails" : {
   "authorizedAmount" : "100",
   "currency" : "jpy"
  }
 },
 "paymentAccountInformation" : {
  "card" : {
   "type" : "001"
  }
 },
 "paymentInformation" : {
  "tokenizedCard" : {
   "type" : "001"
  },
  "card" : {
   "type" : "001"
  }
 },
 "processorInformation" : {
  "salesSlipNumber" : "56311",
  "approvalCode" : "123456",
  "cardVerification" : {
   "resultCode" : "3"
  },
  "responseCategoryCode" : "000",
  "forwardedAcquirerCode" : "Sumitomo",
  "avs" : {
   "code" : "2"
  }
 },
 "reconciliationId" : "002023051806221300000000001",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2023-05-17T21:22:13Z"
}
```

Level II Processing

This section shows you how to process transactions that include Level II data.

Related Information

- See *Level II and Level III Data* on page 47 for a description of and requirements for processing payments that include Level II data.

Additional Resources for Level II/III Payments

For more information, see these guides:

- *Level II and III Processing Developer Guide*
- *API field reference guide for the REST API*
- Github repositories: *https://github.com/Cybersource*

Captures with Level II Data

This section shows you how to capture an authorized transaction with Level II data. These required fields and example are specific to Visa Platform Connect. For required fields, optional fields, and examples specific to your processor see the *Level II and Level III Processing developer guides*.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/captures`
Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures`
The `{id}` is the transaction ID returned in the authorization response.

Required Fields for Capturing a Payment with Level II Data

Use these required fields to capture a payment that includes Level II data.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.invoiceDetails.purchaseOrderNumber	**Required for purchase/procurement cards only.**
orderInformation.invoiceDetails.taxable	**Required if the sum of all orderInformation.lineItems[].taxAmount values > `0`.**
orderInformation.lineItems[].taxAmount	

Related Information

- *API field reference guide for the REST API*

Optional Fields for Capturing a Payment with Level II Data

You can use these optional fields to include additional information when capturing a payment with Level II data.

order.vatTaxAmountSign

orderInformation.shipTo.postalCode

Related Information

- *API field reference guide for the REST API*

REST Example: Capturing a Payment with Level II Data

Request

```
{
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "7.77",
      "currency": "USD"
    },
    "invoiceDetails": {
      "purchaseOrderNumber": "LevelII Auth Po",
      "taxable": true
    },
    "lineItems": [
      {
        "unitPrice": "7.00",
        "taxAmount": ".77"
      }
    ]
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/captures/7316954580096155203955/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/captures/7316954580096155203955"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "id": "7316954580096155203955",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "112.00",
      "currency": "USD"
    }
  },
  "reconciliationId": "7316954318366152603955",
  "status": "PENDING",
  "submitTimeUtc": "2024-11-15T18:30:58Z"
}
```

Credits with Level II Data

This topic shows you how to process a credit with Level II data. These required fields and example are specific to Visa Platform Connect. For required fields, optional fields, and examples specific to your processor see the *Level II and Level III Processing developer guides*.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/credits/`
Test: `POST https://apitest.cybersource.com/pts/v2/credits/`

Required Fields for Processing a Credit with Level II Data

Use these required fields to process a credit that includes Level II data.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine

> whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
orderInformation.invoiceDetails.purchaseOrderNumber	**Required for purchase/procurement cards only.**
orderInformation.invoiceDetails.taxable	**Required if the sum of all orderInformation.lineItems[].taxAmount values > `0`.**
orderInformation.lineItems[].taxAmount	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	

Related Information

- *API field reference guide for the REST API*

Optional Fields for Processing a Credit with Level II Data

You can use these optional fields to include additional information when processing a credit request with Level II data.

order.vatTaxAmountSign

orderInformation.shipTo.postalCode

Related Information

- *API field reference guide for the REST API*

REST Example: Processing a Credit with Level II Data

Request

```
{
  "paymentInformation": {
    "card": {
      "number": "4111111111111111",
      "expirationMonth": "03",
      "expirationYear": "2031"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "7.77",
      "currency": "USD",
      "invoiceDetails": {
        "purchaseOrderNumber": "L2PurchaseOrderNo",
        "purchaseOrderDate": "2024-11-15",
        "taxable": true
      }
    },
    "lineItems": [
      {
        "unitPrice": "7.00",
        "taxAmount": ".77"
      }
    ]
  },
  "billTo": {
    "firstName": "John",
    "lastName": "Deo",
    "address1": "900 Metro Center Blvd",
    "locality": "Foster City",
    "administrativeArea": "CA",
    "postalCode": "48104-2201",
    "country": "US",
    "email": "",
    "phoneNumber": "9321499232"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/credits/7320289208766957204951/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/credits/7320289208766957204951"
    }
  },
  "clientReferenceInformation": {
    "code": "1732028921016"
  },
  "creditAmountDetails": {
    "currency": "USD",
    "creditAmount": "200.00"
  },
  "id": "7320289208766957204951",
  "orderInformation": {
    "amountDetails": {
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "approvalCode": "888888",
    "responseCode": "100"
  },
  "reconciliationId": "69002042N33YJ7AN",
  "status": "PENDING",
  "submitTimeUtc": "2024-11-19T15:08:41Z"
}
```

Sales with Level II Data

This section shows you how to process a sale transaction with Level II data. These required fields and example are specific to Visa Platform Connect. For required fields, optional fields, and examples specific to your processor see the *Level II and Level III Processing developer guides*.
A sale transaction combines and authorization and a capture into a single transaction.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing a Sale with Level II Data

Use these required fields to process a sale that includes Level II data.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
orderInformation.invoiceDetails.purchaseOrderNumber	**Required for purchase/procurement cards only.**
orderInformation.invoiceDetails.taxable	**Required if the sum of all orderInformation.lineItems[].taxAmount values > `0`.**
orderInformation.lineItems[].taxAmount	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
paymentInformation.card.securityCode	**Required only for Visa Platform Connect.**
paymentInformation.card.type	
processingInformation.capture	**Set field to `true`.**

Related Information

- *API field reference guide for the REST API*

Optional Fields for Processing a Sale with Level II Data

You can use these optional fields to include additional information when processing a sale with Level II data.

order.vatTaxAmountSign

orderInformation.shipTo.postalCode

Related Information

- *API field reference guide for the REST API*

REST Example: Processing a Sale with Level II Data

Request

```
{
 "processingInformation": {
  "capture": true
 },
 "orderInformation": {
  "billTo": {
   "country": "US",
   "lastName": "VDP",
   "address1": "201 S. Division St.",
   "postalCode": "48104-2201",
   "locality": "Ann Arbor",
   "administrativeArea": "MI",
   "firstName": "RTS",
   "email": "",
   "invoiceDetails": {
    "purchaseOrderNumber": "L2PurchaseOrderNo",
    "purchaseOrderDate": "2024-11-15",
    "taxable": true
   }
  },
  "amountDetails": {
   "totalAmount": "7.77",
   "currency": "USD"
  },
  "lineItems": [
   {
    "unitPrice": "7.00",
    "taxAmount": ".77"
   }
  ]
 },
 "paymentInformation": {
  "card": {
   "expirationYear": "2031",
   "number": "4111111111111111",
   "expirationMonth": "12",
   "type": "001",
   "securityCode": "999"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
   "href": "/pts/v2/payments/7320306418466109204951/voids"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7320306418466109204951"
  }
 },
 "clientReferenceInformation": {
  "code": "1732030641881"
 },
 "id": "7320306418466109204951",
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "200.00",
   "authorizedAmount": "200.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "processorInformation": {
  "systemTraceAuditNumber": "027986",
  "approvalCode": "831000",
  "cardVerification": {
   "resultCodeRaw": "M",
   "resultCode": "M"
  },
  "merchantAdvice": {
   "code": "01",
   "codeRaw": "M001"
  },
  "responseDetails": "ABC",
  "networkTransactionId": "016153570198200",
  "retrievalReferenceNumber": "432415027986",
  "consumerAuthenticationResponse": {
   "code": "2",
   "codeRaw": "2"
  },
  "transactionId": "016153570198200",
   "responseCode": "00",
   "avs": {
    "code": "Y",
    "codeRaw": "Y"
   }
  },
  "reconciliationId": "73203064184661092O4951",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2024-11-19T15:37:22Z"
 }
```

Level III Processing

This section shows you how to process transactions that include Level III data.

- See *Level II and Level III Data* on page 47 for a description of and requirements for processing payments that include Level III data.

Additional Resources for Level II/III Payments

For more information, see these guides:

- *Level II and III Processing Developer Guide*
- *API field reference guide for the REST API*
- Github repositories: *https://github.com/Cybersource*

Captures with Level III Data

This section shows you how to capture an authorized transaction with Level III data. These required fields and example are specific to Visa Platform Connect. For required fields, optional fields, and examples specific to your processor see the *Level II and Level III Processing developer guides*.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments/{id}/captures`
Test: `POST https://apitest.cybersource.com/pts/v2/payments/{id}/captures`
The `{id}` is the transaction ID returned in the authorization response.

Required Fields for Capturing a Payment with Level III Data

Use these required fields to capture a payment that includes Level III data.

orderInformation.amountDetails.currency

orderInformation.amountDetails.nationalTaxIncluded	Set to `1` if the sum of all orderInformation.lineItems[].taxDetails[].amount values > `0`.
orderInformation.amountDetails.totalAmount	
orderInformation.invoiceDetails.purchaseOrderNumber	Required for purchase/procurement cards only.
orderInformation.invoiceDetails.taxable	Required if the sum of all orderInformation.lineItems[].taxAmount values > `0`.
orderInformation.lineItems[].taxAmount	
processingInformation.purchaseLevel	Set field to `3`.

Related Information

- *API field reference guide for the REST API*

Optional Fields for Capturing a Payment with Level III Data

You can use these optional fields to include additional information when capturing a payment with Level III data.

buyerInformation.vatRegistrationNumber

merchantInformation.cardAcceptorReferenceNumber

merchantInformation.vatRegistrationNumber

order.vatTaxAmountSign

orderInformation.amountDetails.discountAmount

orderInformation.amountDetails.dutyAmount

orderInformation.amountDetails.freightAmount

orderInformation.amountDetails.taxAppliedAfterDiscount

orderInformation.amountDetails.taxAppliedLevel

orderInformation.amountDetails.taxDetails[].amount

orderInformation.amountDetails.taxDetails[].rate

orderInformation.invoiceDetails.commodityCode

orderInformation.invoiceDetails.purchaseContactName

orderInformation.invoiceDetails.purchaseOrderDate

orderInformation.invoiceDetails.vatInvoiceReferenceNumber

orderInformation.lineItems[].commodityCode

orderInformation.lineItems[].discountAmount

orderInformation.lineItems[].discountRate

orderInformation.lineItems[].invoiceNumber

orderInformation.lineItems[].productCode

orderInformation.lineItems[].productName

orderInformation.lineItems[].quantity

orderInformation.lineItems[].taxAppliedAfterDiscount

orderInformation.lineItems[].taxDetails[].amount

orderInformation.lineItems[].taxRate

orderInformation.lineItems[].taxStatusIndicator

orderInformation.lineItems[].totalAmount

orderInformation.lineItems[].typeOfSupply

orderInformation.lineItems[].unitOfMeasure

orderInformation.lineItems[].unitPrice

orderInformation.shippingDetails.shipFromPostalCode

orderInformation.shipTo.administrativeArea

orderInformation.shipTo.postalCode

senderInformation.vatRegistrationNumber

Related Information

- *API field reference guide for the REST API*

REST Example: Capturing a Payment with Level III Data

Request

```
{
  "clientReferenceInformation": {
    "code": "TC50171_14"
  },
  "processingInformation": {
    "purchaseLevel": "3"
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "200.00",
      "currency": "USD"
    },
    "lineItems": [
      {
        "productCode": "service",
        "productName": "TestProduct1",
        "quantity": "2",
        "unitPrice": "40.00",
        "unitOfMeasure": "EA",
        "totalAmount": "100.00",
        "taxAmount": "20.00"
      }
    ],
    "invoiceDetails": {
      "purchaseOrderNumber": "L3PurchaseOrderNo",
      "purchaseOrderDate": "2024-11-15",
      "taxable": true
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/captures/7319475673656287004951/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/captures/7319475673656287004951"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_14"
  },
  "id": "7319475673656287004951",
  "orderInformation": {
    "invoiceDetails": {
      "level3TransmissionStatus": "Y"
    },
    "amountDetails": {
      "totalAmount": "200.00",
      "currency": "USD"
    }
  },
  "reconciliationId": "68960954X33X5WQ2",
  "status": "PENDING",
  "submitTimeUtc": "2024-11-18T16:32:47Z"
}
```

Credits with Level III Data

This topic shows you how to process a credit with Level III data. These required fields and example are specific to Visa Platform Connect. For required fields, optional fields, and examples specific to your processor see the *Level II and Level III Processing developer guides*.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/credits/
Test: POST https://apitest.cybersource.com/pts/v2/credits/

Required Fields for Processing a Credit with Level III Data

Use these required fields to process a credit that includes Level III data.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.nationalTaxIncluded	**Set to 1 if the sum of all orderInformation.lineItems[].taxDetails[].amount values > 0.**
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
orderInformation.invoiceDetails.purchaseOrderNumber	**Required for purchase/procurement cards only.**
orderInformation.invoiceDetails.taxable	**Required if the sum of all orderInformation.lineItems[].taxAmount values > 0.**
orderInformation.lineItems[].taxAmount	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.purchaseLevel	**Set field to 3.**

Related Information

- *API field reference guide for the REST API*

Optional Fields for Processing a Credit with Level III Data

You can use these optional fields to include additional information when processing a credit with Level III data.

buyerInformation.vatRegistrationNumber

merchantInformation.cardAcceptorReferenceNumber

merchantInformation.vatRegistrationNumber

order.vatTaxAmountSign

orderInformation.amountDetails.discountAmount

orderInformation.amountDetails.dutyAmount

orderInformation.amountDetails.freightAmount

orderInformation.amountDetails.taxAppliedAfterDiscount

orderInformation.amountDetails.taxAppliedLevel

orderInformation.amountDetails.taxDetails[].amount

orderInformation.amountDetails.taxDetails[].rate

orderInformation.invoiceDetails.commodityCode

orderInformation.invoiceDetails.purchaseContactName

orderInformation.invoiceDetails.purchaseOrderDate

orderInformation.invoiceDetails.vatInvoiceReferenceNumber

orderInformation.lineItems[].commodityCode

orderInformation.lineItems[].discountAmount

orderInformation.lineItems[].discountRate

orderInformation.lineItems[].invoiceNumber

orderInformation.lineItems[].productCode

orderInformation.lineItems[].productName

orderInformation.lineItems[].quantity

orderInformation.lineItems[].taxAppliedAfterDiscount

orderInformation.lineItems[].taxDetails[].amount

orderInformation.lineItems[].taxRate

orderInformation.lineItems[].taxStatusIndicator

orderInformation.lineItems[].totalAmount

orderInformation.lineItems[].typeOfSupply

orderInformation.lineItems[].unitOfMeasure

orderInformation.lineItems[].unitPrice

orderInformation.shippingDetails.shipFromPostalCode

orderInformation.shipTo.administrativeArea

orderInformation.shipTo.postalCode

senderInformation.vatRegistrationNumber

Related Information

- *API field reference guide for the REST API*

REST Example: Processing a Credit with Level III Data

Request

```
{
  "processingInformation": {
    "purchaseLevel": "3"
  },
  "paymentInformation": {
    "card": {
      "number": "411111111111XXXX",
      "expirationMonth": "03",
      "expirationYear": "2031"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "200",
      "currency": "USD",
      "nationalTaxIncluded": "0",
      "lineItems": [
        {
          "productCode": "service",
          "productName": "TestProduct1",
          "quantity": "2",
          "unitPrice": "40.00",
          "unitOfMeasure": "EA",
          "totalAmount": "100.00",
          "taxAmount": "20.00"
        }
      ],
      "invoiceDetails": {
        "purchaseOrderNumber": "L3PurchaseOrderNo",
        "purchaseOrderDate": "2024-11-15",
        "taxable": true
      }
    },
    "billTo": {
      "firstName": "John",
      "lastName": "Deo",
      "address1": "900 Metro Center Blvd",
      "locality": "Foster City",
      "administrativeArea": "CA",
      "postalCode": "48104-2201",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": "9321499232"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
   "href": "/pts/v2/credits/7320300261536812104951/voids"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/credits/7320300261536812104951"
  }
 },
 "clientReferenceInformation": {
  "code": "1732030026199"
 },
 "creditAmountDetails": {
  "currency": "USD",
  "creditAmount": "200.00"
 },
 "id": "7320300261536812104951",
 "orderInformation": {
  "amountDetails": {
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "reconciliationId": "7320300261536812104951",
 "status": "PENDING",
 "submitTimeUtc": "2024-11-19T15:27:06Z"
}
```

Sales with Level III Data

This section shows you how to process a sale transaction with Level III data. These required fields and example are specific to Visa Platform Connect. For required fields, optional fields, and examples specific to your processor see the *Level II and Level III Processing developer guides*.

A sale transaction combines and authorization and a capture into a single transaction.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing a Sale with Level III Data

Use these required fields to process a sale that includes Level III data.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.nationalTaxIncluded	**Set to `1` if the sum of all orderInformation.lineItems[].taxDetails[].amount values > `0`.**
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
orderInformation.invoiceDetails.purchaseOrderNumber	**Required for purchase/procurement cards only.**
orderInformation.invoiceDetails.taxable	**Required if the sum of all orderInformation.lineItems[].taxAmount values > `0`.**
orderInformation.lineItems[].taxAmount	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	

paymentInformation.card.securityCode	**Required only for Visa Platform Connect.**
paymentInformation.card.type	
processingInformation.capture	**Set field to** `true`**.**
processingInformation.purchaseLevel	**Set field to** `3`**.**

Related Information

- *API field reference guide for the REST API*

Optional Fields for Processing a Sale with Level III Data

You can use these optional fields to include additional information when processing a sale request with Level III data.

buyerInformation.vatRegistrationNumber

merchantInformation.cardAcceptorReferenceNumber

merchantInformation.vatRegistrationNumber

order.vatTaxAmountSign

orderInformation.amountDetails.discountAmount

orderInformation.amountDetails.dutyAmount

orderInformation.amountDetails.freightAmount

orderInformation.amountDetails.taxAppliedAfterDiscount

orderInformation.amountDetails.taxAppliedLevel

orderInformation.amountDetails.taxDetails[].amount

orderInformation.amountDetails.taxDetails[].rate

orderInformation.invoiceDetails.commodityCode

orderInformation.invoiceDetails.purchaseContactName

orderInformation.invoiceDetails.purchaseOrderDate

orderInformation.invoiceDetails.vatInvoiceReferenceNumber

orderInformation.lineItems[].commodityCode

orderInformation.lineItems[].discountAmount

orderInformation.lineItems[].discountRate

orderInformation.lineItems[].invoiceNumber

orderInformation.lineItems[].productCode

orderInformation.lineItems[].productName

orderInformation.lineItems[].quantity

orderInformation.lineItems[].taxAppliedAfterDiscount

orderInformation.lineItems[].taxDetails[].amount

orderInformation.lineItems[].taxRate

orderInformation.lineItems[].taxStatusIndicator

orderInformation.lineItems[].totalAmount

orderInformation.lineItems[].typeOfSupply

orderInformation.lineItems[].unitOfMeasure

orderInformation.lineItems[].unitPrice

orderInformation.shippingDetails.shipFromPostalCode

orderInformation.shipTo.administrativeArea

orderInformation.shipTo.postalCode

senderInformation.vatRegistrationNumber

Related Information

- *API field reference guide for the REST API*

REST Example: Processing a Sale with Level III Data

Request

```
{
 "processingInformation": {
  "capture": true,
  "purchaseLevel": "3"
 },
 "orderInformation" : {
  "billTo" : {
  "country" : "US",
  "lastName" : "VDP",
  "address1" : "201 S. Division St.",
  "postalCode" : "48104-2201",
  "locality" : "Ann Arbor",
  "administrativeArea" : "MI",
  "firstName" : "RTS",
  "email" : "test@cybs.com",
   "lineItems": [
   {
    "productCode": "service",
    "productName": "TestProduct1",
    "quantity": "2",
    "unitPrice": "40.00",
    "unitOfMeasure": "EA",
    "totalAmount": "100.00",
    "taxAmount": "20.00"
   }
  ],
  "invoiceDetails": {
   "purchaseOrderNumber": "L3PurchaseOrderNo",
   "purchaseOrderDate": "2024-11-15",
   "taxable": true
  }
 },
  "amountDetails" : {
   "totalAmount" : "200.00",
   "currency" : "USD"
  }
 },
 "paymentInformation" : {
  "card" : {
   "expirationYear" : "2031",
   "number" : "4111111111111111",
   "expirationMonth" : "12",
   "type" : "001",
   "securityCode": "999"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
   "href": "/pts/v2/payments/7320309047086907404953/voids"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7320309047086907404953"
  }
 },
 "clientReferenceInformation": {
  "code": "1732030904750"
 },
 "id": "7320309047086907404953",
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "200.00",
   "authorizedAmount": "200.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "processorInformation": {
  "systemTraceAuditNumber": "528726",
  "approvalCode": "831000",
  "cardVerification": {
   "resultCodeRaw": "M",
   "resultCode": "M"
  },
  "merchantAdvice": {
   "code": "01",
   "codeRaw": "M001"
  },
  "responseDetails": "ABC",
  "networkTransactionId": "016153570198200",
  "retrievalReferenceNumber": "432415528726",
  "consumerAuthenticationResponse": {
   "code": "2",
   "codeRaw": "2"
  },
  "transactionId": "016153570198200",
  "responseCode": "00",
  "avs": {
   "code": "Y",
   "codeRaw": "Y"
  }
 },
 "reconciliationId": "7320309047086907404953",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2024-11-19T15:41:44Z"
}
```

Mastercard Processing

These use cases are specific to Mastercard processing.

Mastercard Bill Payment Processing

This section describes how to request an authorization for a Mastercard Bill Payment.

Field Specific to this Use Case

Include this field with a standard authorization request when processing a Mastercard Bill Payment:

processingInformation.authorizationOptions.billPaymentType

Set the value to indicate the type of bill that the cardholder is paying.

Requirements

Sign up with Mastercard to participate in their bill payment program.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Related Information

- See *Mastercard Bill Payments* on page 48 for a description of and requirements for processing Mastercard Bill Payments.

Required Fields for Authorizing a Mastercard Bill Payment

Use these required fields to authorize a Mastercard bill payment.

Important

> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

processingInformation.authorizationOptions.billPaymentType

Set the value to indicate the type of bill that the cardholder is paying.

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a Mastercard Bill Payment

Request

```
{
  "orderInformation": {
    "billTo": {
      "country": "BR",
      "lastName": "Doe",
      "firstName": "John",
      "address1": "Av Pres Juscelino Kubistchek 1909",
      "address2": "",
      "postalCode": "04543907",
      "locality": "Sao Paulo",
      "administrativeArea": "SP",
      "email": "john.doe@company.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
            "currency": "BRL"
        }
    },
    "paymentInformation": {
        "card": {
            "expirationMonth": "12",
            "expirationYear": "2031",
            "number": "555555555555xxxx",
            "securityCode": "123",
            "type": "002"
        }
    },
    "processingInformation": {
        "authorizationOptions": {
            "billPaymentType": "001"
        }
    }
}
```

Response to a Successful Request

```
{
  "_links" : {
    "authReversal" : {
      "method" : "POST",
      "href" : "/pts/v2/payments/6863356803746501803955/reversals"
    },
    "self" : {
      "method" : "GET",
      "href" : "/pts/v2/payments/6863356803746501803955"
    },
    "capture" : {
      "method" : "POST",
      "href" : "/pts/v2/payments/6863356803746501803955/captures"
    }
  },
  "clientReferenceInformation" : {
    "code" : "1686335680358"
  },
  "id" : "6863356803746501803955",
  "orderInformation" : {
    "amountDetails" : {
      "authorizedAmount" : "100.00",
      "currency" : "brl"
    }
  },
  "paymentAccountInformation" : {
    "card" : {
      "type" : "002"
    }
  },
  "paymentInformation" : {
    "tokenizedCard" : {
      "type" : "002"
    },
    "card" : {
      "type" : "002"
    }
  },
  "processorInformation" : {
    "approvalCode" : "010012",
    "networkTransactionId" : "999010012",
    "transactionId" : "72b2900a9f316142b627a21031b48b0c259f08ffba0004172a04450c5d212345",
    "responseCode" : "400",
    "avs" : {
      "code" : "2"
    }
  },
  "reconciliationId" : "NHRRGOVtUxkb",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2023-06-09T18:34:40Z"
}
```

Mastercard Expert Monitoring Solutions Processing

This section shows you how to obtain the transaction fraud score assigned by Mastercard Expert Monitoring Solutions.

Requirement

Contact customer support to enable Mastercard Expert Monitoring Solutions for your account.

> **Important**
>
> After this feature is enabled for your account, Mastercard returns a fraud score for all your card-not-present authorization requests for Mastercard payment cards issued in the US.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Related Information

- See *Mastercard Expert Monitoring Solutions* on page 48 for a description of the transaction fraud score determined by Mastercard Expert Monitoring Solutions.

Required Fields for Processing an Authorization with Mastercard Expert Monitoring Solutions

Use these required fields to process an authorization using Mastercard Expert Monitoring Solutions.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

Related Information

- *API field reference guide for the REST API*

Response Field for Authorizations with Mastercard Expert Monitoring Solutions

This field can be returned in a response to an authorization using Mastercard Expert Monitoring Solutions.

processorInformation.emsTransactionRiskScore **Fraud score for a Mastercard transaction.**

Related Information

- *API field reference guide for the REST API*

REST Example: Obtaining the Mastercard Fraud Score for an Authorization

Request

```
{
  "orderInformation": {
    "billTo": {
      "country": "US",
      "lastName": "Kim",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "firstName": "Kyong-Jin",
      "email": "kim.test@cybs.com"/>"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "usd"
    }
  },
  "paymentInformation": {
    "card": {
      "number": "555555555555xxxx",
      "expirationYear": "2031",
      "expirationMonth": "12",
      "type": "002"
    }
  }
}
```

Response to a Successful Request

The **processorInformation.emsTransactionRiskScore** response field contains the fraud score returned by Mastercard Expert Monitoring Solutions. In this example, the fraud score indicates a high likelihood (field value `843`) of suspicious service station activity (field value `09`).

```
{
 "_links" : {
  "authReversal" : {
   "method" : "POST",
   "href" : "/pts/v2/payments/6461731521426399003473/reversals"
  },
  "self" : {
   "method" : "GET",
   "href" : "/pts/v2/payments/6461731521426399003473"
  },
  "capture" : {
   "method" : "POST",
      "href" : "/pts/v2/payments/6461731521426399003473/captures"
    }
  },
  "clientReferenceInformation" : {
    "code" : "1646173152047"
  },
  "id" : "6461731521426399003473",
  "orderInformation" : {
    "amountDetails" : {
      "authorizedAmount" : "100.00",
      "currency" : "usd"
    }
  },
  "paymentAccountInformation" : {
    "card" : {
      "type" : "002"
    }
  },
  "paymentInformation" : {
    "tokenizedCard" : {
      "type" : "002"
    },
    "card" : {
      "type" : "002"
    }
  },
 "paymentInsightsInformation" : {
    "responseInsights" : {
      "categoryCode" : "01"
    }
  },
  "processorInformation" : {
    "emsTransactionRiskScore": "84309",
    "systemTraceAuditNumber" : "862481",
    "approvalCode" : "831000",
    "merchantAdvice" : {
      "code" : "01",
      "codeRaw" : "M001"
    },
    "responseDetails" : "ABC",
    "networkTransactionId" : "016153570198200",
    "consumerAuthenticationResponse" : {
      "code" : "2",
      "codeRaw" : "2"
    },
    "transactionId" : "016153570198200",
    "responseCode" : "00",
    "avs" : {
      "code" : "Y",
      "codeRaw" : "Y"
    }
  },
  "reconciliationId" : "6461731521426399003473",
  "status" : "AUTHORIZED",
  "submitTimeUtc" : "2023-06-09T22:19:12Z"
}
```

Payer Authentication Processing

This section shows you how to process authorizations that use these payer authentication methods:

- American Express: SafeKey
- JCB: J/Secure
- Mastercard: Identity Check
- Visa: Visa Secure

Related Information

- See the *Payer Authentication Developer Guide* for details about payer authentication.

Additional Resources for Payer Authentication

For more information, see these guides:

- *Payer Authentication Developer Guide*
- *API field reference guide for the REST API*
- Github repositories: *https://github.com/Cybersource*

Providing Payer Authentication Information for Authorization

The values that are returned from payer authentication must be provided when seeking authorization for the transaction. Authentication information that is not included when

considering authorization may cause the transaction to be refused or downgraded and prevent the normal liability shift from occurring.
The level of security in payer authentication is denoted by the two digit Electronic Commerce Indicator (ECI) that is assigned to the transaction. These digital values have text equivalents which are assigned to the **processingInformation.commerceIndicator** field.

The American Express, Diners, Discover, UPI, and Visa card brands use 05, 06, and 07 digit values to express the authentication level for a 3-D Secure transaction.

Text Values for ECI Values

ECI Value	Meaning	Visa	Diners	Discover	UPI	Amex
05	Authenticated	vbv	pb	dipb	up3ds	aesk
06	Attempted authentication with a cryptogram	vbv_attempted	pb_attempted	dipb_attempted	up3ds_attempted	aesk_attempted
07	Internet, not authenticated	vbv_failure/internet	internet	internet	up3ds_failure/internet	internet

Mastercard and Maestro cards use 00, 01, 02, 06, and 07 digit values to indicate the authentication level of the transaction.

Mastercard/Maestro Text Values for ECI Values

ECI Value	Meaning	Mastercard/Maestro
00	Internet, not authenticated	spa/internet
01	Attempted authentication	spa
02	Authenticated	spa
06	Exemption from authentication or network token without 3#D Secure	spa
07	Authenticated merchant-initiated transaction	spa

The payer authentication response contains other information that needs to be passed on for successful authorization. Be sure to include these fields when requesting a separate authorization:

- **consumerAuthenticationInformation.directoryServerTransactionId** (Mastercard, Maestro, UPI only)
- **consumerAuthenticationInformation.eciRaw**

- **consumerAuthenticationInformation.paresStatus**
- **consumerAuthenticationInformation.paSpecificationVersion**
- **consumerAuthenticationInformation.ucafAuthenticationData** (Mastercard/Maestro only)
- **consumerAuthenticationInformation.ucafCollectionIndicator** (Mastercard/Maestro only)
- **consumerAuthenticationInformation.cavv**
- **consumerAuthenticationInformation.xid**

American Express SafeKey

American Express SafeKey is the authentication service in the American Express card network that uses the 3-D Secure protocol to validate customers at checkout. When you request an authorization using a supported card type and a supported processor, you can include payer authentication data in the request.

Before implementing payer authentication for American Express SafeKey, contact customer support to have your account configured for this feature.

Fields Specific to the American Express SafeKey Use Case

These API fields are required specifically for this use case.

consumerAuthenticationInformation.cavv	Required when payer authentication is successful.
processingInformation.commerceIndicator	Set this field to one of these values: • `aesk`: Successful authentication (3-D Secure value of `05`). • `aesk_attempted`: Authentication was attempted (3-D Secure value of `06`). • `internet`: Authentication failed or was not attempted (3-D Secure value of `07`).

Processor-Specific Requirements

Visa Platform Connect

processingInformation.authorizationOptions.transaction	Required only for merchants in Saudi Arabia.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`

Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Related Information

- *Payer Authentication Developer Guide | REST API*

Required Fields for Processing an Authorization Using American Express SafeKey

These fields must be included in a request for an authorization with American SafeKey. The values for these fields are in the response from the payer authentication validate service. When you request the payer authentication validate and authorization services together, the data is automatically passed from one service to the other.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

clientReferenceInformation.code

consumerAuthenticationInformation.cavv

consumerAuthenticationInformation.eciRaw Required when the payer authentication validation service returns a raw unmapped ECI value.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

paymentInformation.card.type

processingInformation.commerceIndicator Set this field to one of these values:

- aesk: Successful authentication (3-D Secure value of 05).
- aesk_attempted: Authentication was attempted (3-D Secure value of 06).
- internet: Authentication failed or was not attempted (3-D Secure value of 07).

Related Information

- *API field reference guide for the REST API*

Optional Field for Processing an Authorization Using American Express SafeKey

This field is optional in a request for an authorization with American Express SafeKey. The value for this field is in the response from the payer authentication validate service. When you request the payer authentication validate and authorization services together, the data is automatically passed from one service to the other.

consumerAuthenticationInformation.xid

Related Information

- *API field reference guide for the REST API*

REST Example: Processing an Authorization Using American Express SafeKey

Request

```
{
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "processingInformation": {
     "commerceIndicator": "aesk"
  },
  "paymentInformation": {
     "card": {
     "number": "3400000XXXXXXX8",
     "expirationMonth": "01",
     "expirationYear": "2025"
   }
 },
   "orderInformation": {
     "amountDetails": {
     "totalAmount": "100",
     "currency": "USD"
   },
   "billTo": {
    "firstName": "John",
    "lastName": "Smith",
    "address1": "201 S. Division St._1",
    "locality": "Foster City",
    "administrativeArea": "CA",
    "postalCode": "94404",
    "country": "US",
    "email": "accept@who.com",
    "phoneNumber": "6504327113"
   }
  },
    "consumerAuthenticationInformation": {
    "cavv": "1234567890987654321ABCDEFabcdefABCDEF123",
    "xid": "1234567890987654321ABCDEFabcdefABCDEF123"
    }
  }
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6783071542936193303955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6783071542936193303955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6783071542936193303955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "id": "6783071542936193303955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "100.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "003"
  }
 },
 "paymentInformation": {
  "accountFeatures": {
   "currency": "usd",
   "balanceAmount": "70.00"
  },
  "tokenizedCard": {
   "type": "003"
```

```
    },
    "card": {
      "type": "003"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62427259FEYR18Q2",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-03-08T20:25:54Z"
}
```

JCB J/Secure

JCB J/Secure is the authentication service in the JCB card network that uses the 3-D Secure protocol to validate customers at checkout. When you request an authorization using a supported card type and a supported processor, you can include payer authentication data in the request. The payer authentication services enable you to add payer authentication support to your website without running additional software on your server.

Before implementing payer authentication for JCB J/Secure, contact customer support to have your account configured for this feature.

Fields Specific to the JCB J/Secure Use Case

These API fields are required specifically for this use case.

consumerAuthenticationInformation.cavv
Required when payer authentication is successful.

consumerAuthenticationInformation.xid
Required when payer authentication is successful.

consumerAuthenticationInformation.eciRaw
Required when the payer authentication validation service returns a raw ECI value.

processingInformation.commerceIndicator
Set this field to one of these values:

- `js`: Successful authentication for a JCB card (3-D Secure value of `05`).

- `js_attempted`: Authentication was attempted for a JCB card (3-D Secure value of `06`).
- `js_failure`: or `internet`: Authentication failed or was not attempted for a JCB card (3-D Secure value of `07`).

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Related Information

- *Payer Authentication Developer Guide | REST API*

Required Fields for Processing an Authorization Using JCB J/Secure Authentication

Use these required fields to process an authorization using JCB J/Secure authentication.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

clientReferenceInformation.code

consumerAuthenticationInformation.cavv

consumerAuthenticationInformation.xid

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
paymentInformation.card.type	
processingInformation.commerceIndicator	Set this field to one of these values: • `js`: Successful authentication (3-D Secure value of `05`). • `js_attempted`: Authentication was attempted (3-D Secure value of `06`). • `js_failure`: Authentication failed or was not attempted (3-D Secure value of `07`).

Related Information

- *API field reference guide for the REST API*

REST Example: Processing an Authorization Using JCB J/Secure Authentication

Request

```
{
 "clientReferenceInformation": {
   "code": "TC50171_3"
 },
 "processingInformation": {
    "commerceIndicator": "js"
 },
 "paymentInformation": {
    "card": {
    "number": "3400000XXXXXXX8",
    "expirationMonth": "01",
    "expirationYear": "2025"
  }
},
  "orderInformation": {
    "amountDetails": {
    "totalAmount": "100",
    "currency": "USD"
  },
  "billTo": {
    "firstName": "John",
    "lastName": "Smith",
    "address1": "201 S. Division St._1",
    "locality": "Foster City",
    "administrativeArea": "CA",
    "postalCode": "94404",
    "country": "US",
    "email": "accept@who.com",
    "phoneNumber": "6504327113"
   }
  },
    "consumerAuthenticationInformation": {
    "cavv": "1234567890987654321ABCDEFabcdefABCDEF123",
    "xid": "1234567890987654321ABCDEFabcdefABCDEF123"
    }
  }
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6783071542936193303955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6783071542936193303955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6783071542936193303955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "id": "6783071542936193303955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "100.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "003"
  }
 },
 "paymentInformation": {
  "accountFeatures": {
   "currency": "usd",
   "balanceAmount": "70.00"
  },
  "tokenizedCard": {
   "type": "003"
  },
  "card": {
   "type": "003"
  }
 },
 "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62427259FEYR18Q2",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-03-08T20:25:54Z"
}
```

Mastercard Identity Check

Mastercard Identity Check is the authentication service in the Mastercard card network that uses the 3-D Secure protocol in online transactions to authenticate customers at checkout.

Mastercard Identity Check generates a unique, 32-character transaction token, called the account authentication value (AAV) each time a Mastercard Identity Check-enabled account holder makes an online purchase. The AAV binds the account holder to a specific transaction. Mastercard Identity Check transactions use the universal cardholder authentication field (UCAF) as a standard to collect and pass AAV data.

Before implementing payer authentication for Mastercard Identity Check, contact customer support to have your account configured for this feature.

Fields Specific to the Mastercard Identity Check Use Case

These API fields are required specifically for this use case.

consumerAuthenticationInformation. directory ServerTransactionId	Set this field to the transaction ID returned by Mastercard Identity Check during the authentication process.
consumerAuthenticationInformation. paSpecificationVersion	Set this field to the Mastercard Identity Check version returned by Mastercard Identity Check during the authentication process.
consumerAuthenticationInformation. ucafCollectionIndicator	Set to the last digit of the raw ECI value returned from authentication. For example, if ECI=02, this value should be 2.
processingInformation.commerceIndicator	Set this field to one of these values: • spa: Successful authentication (3-D Secure value of 02).

- spa: Authentication was attempted (3-D Secure value of 01).
- spa or internet: Authentication failed or was not attempted (3-D Secure value of 00)

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for Processing an Authorization Using Mastercard Identity Check

Use these required fields to process an authorization using Mastercard Identity Check.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

consumerAuthenticationInformation.directoryServerTransactionId

consumerAuthenticationInformation.paSpecificationVersion

consumerAuthenticationInformation.ucafCollectionIndicator

Set to the last digit of the raw ECI value returned from authentication. For example, if ECI=02, this value should be 2.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.commerceIndicator	**Set this field to one of these values:** • spa: Successful authentication (3-D Secure value of 02). • spa: Authentication was attempted (3-D Secure value of 01). • spa or internet: Authentication failed or was not attempted (3-D Secure value of 00)

Related Information

- *API field reference guide for the REST API*

REST Example: Processing an Authorization Using Mastercard Identity Check

Request

```
{
  "clientReferenceInformation" : {
    "code" : "TC50171_6"
  },
  "consumerAuthenticationInformation" : {
    "ucafCollectionIndicator" : "2",
    "ucafAuthenticationData" : "EHuWW9PiBkWvqE5juRwDzAUFBAk",
    "directoryServerTransactionId" : "f38e6948-5388-41a6-bca4-b49723c19437",
    "paSpecificationVersion" : "2.2.0"
  },
  "processingInformation" : {
    "commerceIndicator" : "spa"
    },
    "orderInformation" : {
      "billTo" : {
        "country" : "US",
        "lastName" : "Deo",
        "address1" : "201 S. Division St.",
        "postalCode" : "48104-2201",
        "locality" : "Ann Arbor",
        "administrativeArea" : "MI",
        "firstName" : "John",
        "email" : test@cybs.com
      },
      "amountDetails" : {
        "totalAmount" : "105.00",
        "currency" : "USD"
      }
    },
    "paymentInformation" : {
```

```
   "card" : {
    "expirationYear" : "2031",
    "number" : "555555555555XXXX",
    "securityCode" : "123",
    "expirationMonth" : "12",
    "type" : "002"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6758990751436655004951/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6758990751436655004951"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6758990751436655004951/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "id": "6758990751436655004951",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "100.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "002"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "002"
  },
  "card": {
   "type": "002"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "authIndicator": "1",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "71183995FDU0YRTK",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-02-08T23:31:15Z"
}
```

Visa Secure

Visa Secure is the authentication service in the Visa card network that uses the 3-D Secure protocol to authenticate customers at checkout. This authentication is a two-step process. First, the cardholder is authenticated by 3-D Secure. Then, the transaction is authorized based on the 3-D Secure evaluation. This section explains how to authorize a card payment based on the 3-D Secure evaluation.

Before implementing Visa Secure, contact customer support to have your account configured for this feature.

Fields Specific to the Visa Secure Use Case

These API fields are required specifically for this use case.

processingInformation.commerceIndicator	Set the value to vbv for a successful authentication (3-D Secure value of 05), vbv_attempted if authentication was attempted but did not succeed (3-D Secure value of 06), or vbv_failure if authentication failed (3-D Secure value of 07).
consumerAuthenticationInformation.cavv	Required when payer authentication is successful.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments

Test: POST https://apitest.cybersource.com/pts/v2/payments

Related Information

- *API field reference guide for the REST API*

Required Fields for Processing an Authorization Using Visa Secure

Use these required fields to process an authorization using Visa Secure.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. Refer to the Payments guide for more information about relaxed requirements in payment transactions.

Required Fields

clientReferenceInformation.code

consumerAuthenticationInformation.cavv

This field is required when payer authentication is successful. Otherwise, this field is optional.

consumerAuthenticationInformation.xid

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

paymentInformation.card.type

processingInformation.commerceIndicator

Set this field to one of these values:

- `vbv`: Successful authentication (EMV 3-D Secure value of `05`).
- `vbv_attempted`: Authentication was attempted (EMV 3-D Securevalue of `06`).
- `vbv_failure`: or `internet`: Authentication failed or was not attempted (EMV 3-D Secure value of `07`).

Related Information

- *API field reference guide for the REST API*

REST Example: Validating and Authorizing a Transaction

Request

```
{
  "clientReferenceInformation": {
    "code": "test"
  },
  "processingInformation": {
    "capture": "true",
    "authorizationOptions": {
      "ignoreAvsResult": "true"
    },
    "actionList": [
      "VALIDATE_CONSUMER_AUTHENTICATION"
    ]
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4000000000002503",
      "securityCode": "123",
      "expirationMonth": "12",
      "type": "001"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "GBP"
    },
    "billTo": {
      "firstName": "John",
      "lastName": "Smith",
      "address1": "201 S. Division St._1",
      "address2": "Suite 500",
      "locality": "Foster City",
      "administrativeArea": "CA",
      "postalCode": "94404",
      "country": "US",
      "email": "accept@cybersource.com",
      "phoneNumber": "6504327113"
    }
  },
  "consumerAuthenticationInformation": {
    "authenticationTransactionId": "2b4eAa4K3H778X34Ciy0"
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/payments/7478305945626990404807/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/7478305945626990404807"
    }
  },
  "clientReferenceInformation": {
    "code": "test"
  },
  "consumerAuthenticationInformation": {
    "indicator": "vbv",
    "eciRaw": "05",
    "authenticationResult": "0",
    "strongAuthentication": {
      "OutageExemptionIndicator": "0"
    },
    "authenticationStatusMsg": "Success",
    "eci": "05",
    "token": "Axj//wSTlWZX08jkcOTHAAIU3YMmzhgzcN2ie/LXsgSgKe/LXsgS500OnEFBWGTSTL0Yua1eAwHScqzK
+nkcjhyY4wDi0",
    "cavv": "AAIBBYNoEwAAACcKhAJkdQAAAAA=",
    "paresStatus": "Y",
    "xid": "AAIBBYNoEwAAACcKhAJkdQAAAAA=",
    "directoryServerTransactionId": "fa628ed8-ad77-4723-b28f-91952eaca8fe",
    "threeDSServerTransactionId": "71399671-8456-4c97-b056-e127622a5e26",
    "specificationVersion": "2.2.0",
    "acsTransactionId": "5f9fb589-08cc-4952-866d-30939868f411"
  },
  "id": "7478305945626990404807",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "authorizedAmount": "100.00",
      "currency": "GBP"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "brandName": "VISA",
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "bin": "400000",
      "type": "VISA"
    }
  },
  "pointOfSaleInformation": {
        "terminalId": "12345678"
    },
    "processorInformation": {
        "paymentAccountReferenceNumber": "V0010013018036776997406844475",
        "merchantNumber": "12345678",
        "approvalCode": "100",
        "cardVerification": {
            "resultCodeRaw": "3",
            "resultCode": "2"
        },
        "merchantAdvice": {
            "code": "00",
            "codeRaw": "0"
        },
        "networkTransactionId": "123456789012345",
        "transactionId": "123456789012345",
        "responseCode": "0",
        "avs": {
            "code": "U",
            "codeRaw": "00"
        }
    },
    "reconciliationId": "7026803874",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2025-05-21T12:29:54Z"
}
```

Relaxed Requirements for Address Data and Expiration Date in Payment Transactions

With relaxed requirements for address data and the expiration date, not all standard payment request fields are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required.

Requirements

You must contact customer support in order to enable relaxed requirements for address data and expiration date.

Services

Relaxed requirements for address data and expiration date are supported for these services:

- Authorization
- Capture
- Stand-alone credit
- Subscription create
- Subscription update

Relaxed Fields

> **Important**
>
> When relaxed requirements for address data and expiration date are enabled for your Cybersource account, and your service request does not include one or more of the fields in the following list, you increase the risk of declined transactions and fraud depending on your location, your processor, and the cardholder's issuing bank.

It is your responsibility to determine whether a field is required for the transaction you are requesting. For example, an issuing bank can decline an authorization request for a recurring transaction with a Visa Europe card if the expiration date is incorrect, invalid, or missing. If you do not provide the correct expiration date for a recurring transaction the authorization request may be declined.

orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	When you include this field in your request, you must also include paymentInformation.card.expirationYear. You can submit an expiration date that has expired. This exception does not apply when you combine any of the services listed above with any other service. This field is required for payment network token transactions and subscription creation requests.
paymentInformation.card.expirationYear	When you include this field in your request, you must also include paymentInformation.card.expirationMonth. You can submit an expiration date that has expired. This exception does not apply

when you combine any of the services listed above with any other service.

This field is required for payment network token transactions and subscription creation requests.

Split Shipments Processing

Split shipments enable you to split an order into multiple shipments with multiple captures. You can use this feature when a customer orders a product that is not yet available.

> **Important**
>
> Split shipments are not available for Mastercard transactions in the IDR currency on Visa Platform Connect.

Multiple partial captures and split shipments are not the same feature. The processor provides the multiple partial captures feature, while Cybersource provides the split shipment feature.

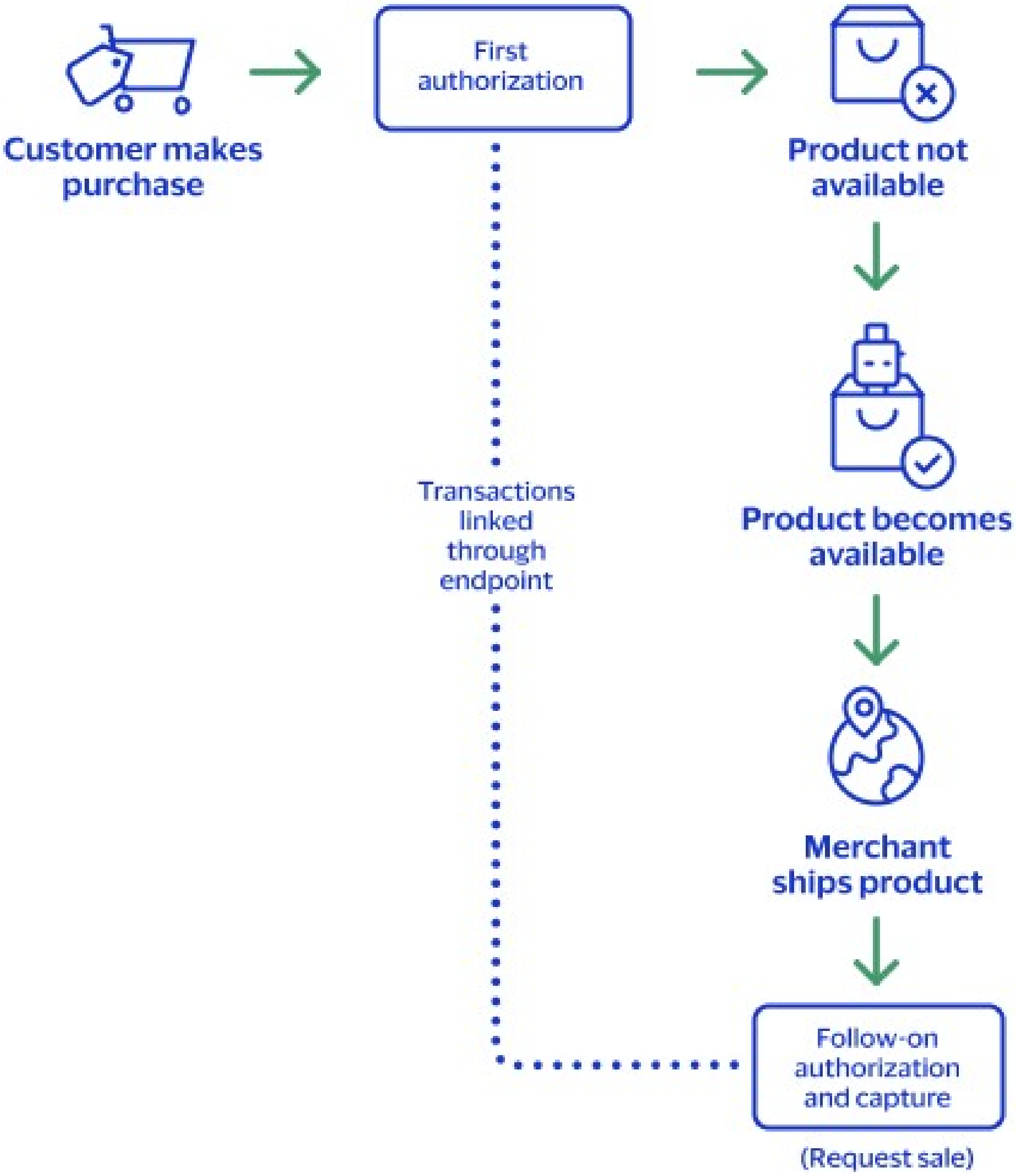
Requirements for Using Split Shipments

The requirements for using split shipments are you must use Visa Platform Connect and contact customer support to have your account configured for this feature.

> **Important**
>
> A Visa Platform Connect account can only be enabled for either the multiple partial captures or split shipments feature, but not both.

Authorizing a Sale for a Product Not Yet Available

When the customer purchases a product that is not yet available, you can request an authorization and a sale. First request an authorization to ensure that funds are available. After the product becomes available, ship the product and request a sale. Cybersource then links the follow-on authorization to the first authorization, and then links to the capture request.



Authorizing a Sale for a Product not yet Available

Step 1: Requesting an authorization
Request an authorization to ensure that funds are available before the product is available for immediate shipment. The authorization request requires no additional fields or requirements than a basic authorization.
Step 2: Processing a sale
When the product becomes available, ship the product and request a sale. The follow-on authorization requires you to submit a sale request that includes the

processingInformation.linkId field in addition to the basic fields required for every sale request. The **processingInformation.linkId** field in an authorization request triggers the split-shipment functionality.
Set the **processingInformation.linkId** field to the **{id}** value from the endpoint.
Field Specific to authorizing a sale for a product not yet available:
First Authorization Response: The **{id}** value is returned in the endpoint.
Follow-on Authorization Request: `processingInformation.linkId=SWVdPS5IM`
Step 3: Cybersource attempts to link the follow-on authorization request to the first authorization

- If the **processingInformation.linkId** value is valid, the follow-on authorization is linked to the original authorization in the Business Center and in reports.
- If the **processingInformation.linkId** value is not valid, the follow-on authorization is not linked to the original authorization in the Business Center and in reports.

Step 4: Cybersource links the capture request

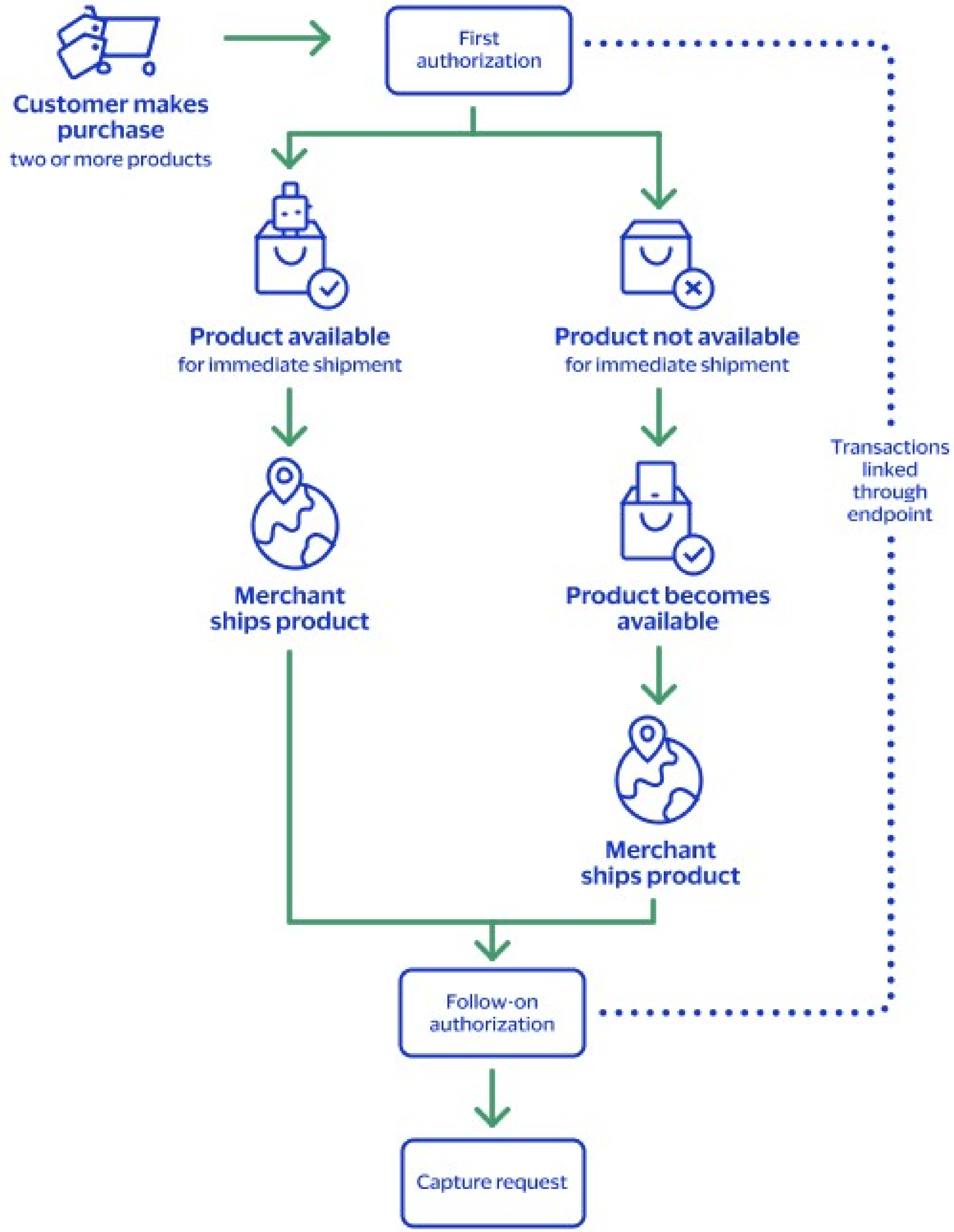
- If the **processingInformation.linkId** value for the follow-on authorization was valid, all three transactions (first authorization, follow-on authorization, capture) are linked together in the Business Center and in reports.
- If the **processingInformation.linkId** value for the follow-on authorization was not valid, the second authorization and capture are linked to each other in the Business Center and in reports, but they are not linked to the first authorization.

Related Information

- See *Basic Authorizations* on page 58 for information on how to process a basic authorization.
- See *Sale* on page 105 for information on how to process a sale.

Processing Two Authorizations and a Capture for Multiple Products

When the customer purchases a product that is not yet available, you can request two authorizations and a capture. First request an authorization to ensure that funds are available, and then ship the available products. After the remaining products become available, request follow-on authorization to ensure funds are still available. Ship the remaining products, and request a capture. Cybersource links the follow-on authorization to the first authorization and the capture request to the other transactions.



Processing Two Authorizations and a Capture for Multiple Products

Step 1: Requesting an authorization

Request an authorization to ensure that funds are available for one or more of the products that are available for immediate shipment. The authorization request requires no additional fields or requirements than a basic authorization.

Step 2: Requesting a follow-on authorization

After the product becomes available, request a follow-on authorization to ensure that funds are still available. The follow-on authorization request must include the **processingInformation.linkId** field in addition to the basic fields required for every authorization request. The **processingInformation.linkId** field in an authorization request triggers the split shipment functionality.

Set the **processingInformation.linkId** field to the **{id}** value from the endpoint.

Field specific to requesting a follow-on authorization request:

First Authorization Response: The **{id}** value is returned in the endpoint.

Follow-on Authorization Request: `processingInformation.linkId=SWVdPS5IM`

Step 3: Cybersource attempts to link the follow-on authorization request to the first authorization

- If the **processingInformation.linkId** value is valid, the follow-on authorization is linked to the original authorization in the Business Center and in reports.
- If the **processingInformation.linkId** value is not valid, the follow-on authorization is not linked to the original authorization in the Business Center and in reports.

Step 4: Requesting a capture

You ship the product and request a capture. The capture request requires only the basic fields as any capture request.

Step 5: Cybersource attempts to link the capture request to the other transactions

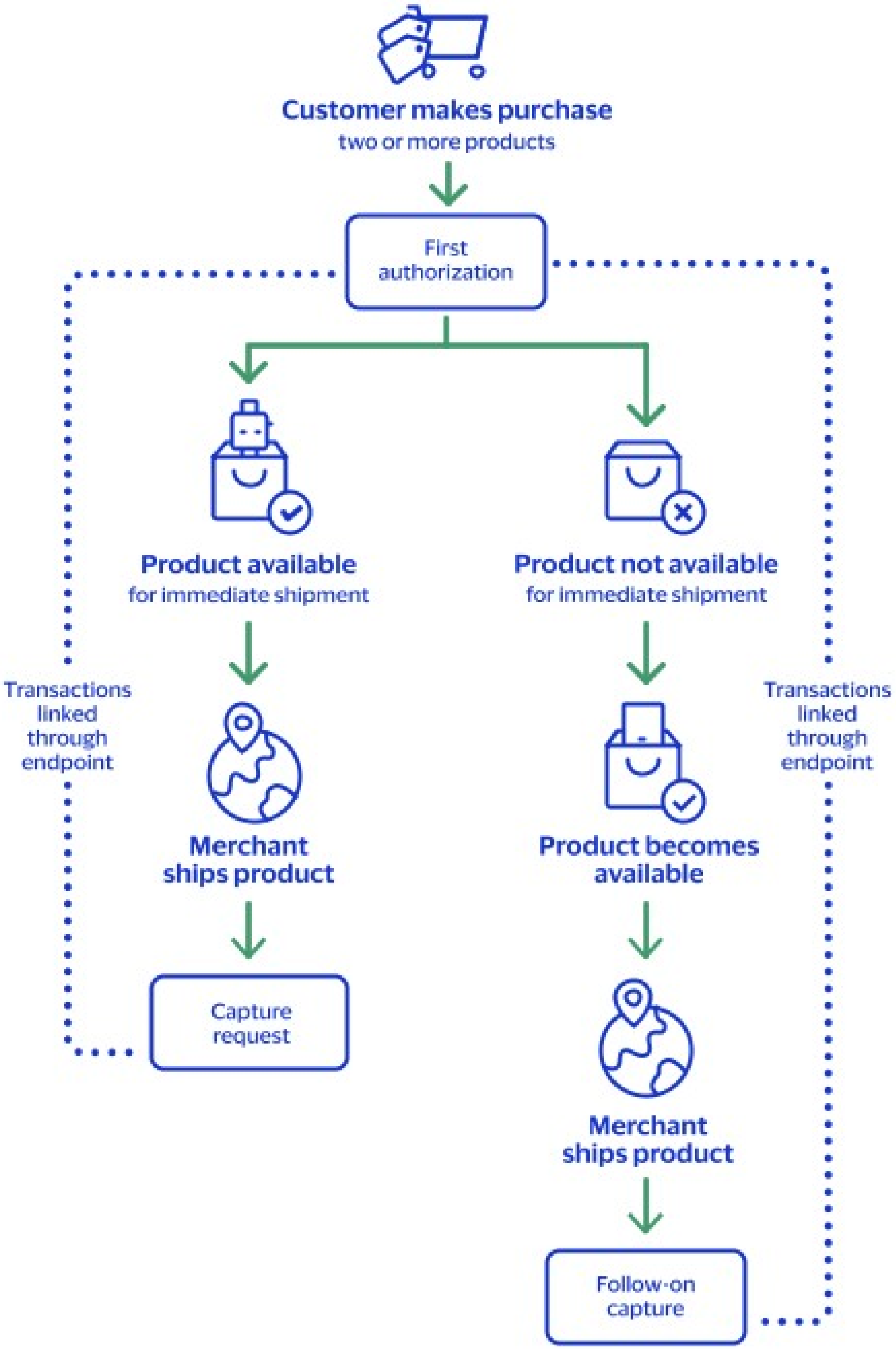
All three transactions (first authorization, follow-on authorization, capture) are linked together in the Business Center and in reports.

Related Information

- See *Basic Authorizations* on page 58 for information on how to process a basic authorization.
- See *Captures* on page 112 for information on how to process a capture.

Processing an Authorization and Two Captures for Multiple Products

When the customer orders multiple products and one is not available, you must request an authorization to ensure funds are available. You ship the products that are available and request a capture for the amount of the shipped products. When the remaining product becomes available, ship the product and request a follow-on capture for the amount of the product. Cybersource performs a system-generated authorization for the follow-on capture request. Cybersource then links the capture request. You receive the status of the follow-on capture request and its associated system-generated authorization.



Processing an Authorization and Two Captures for Multiple Products

Step 1: Requesting an authorization

Request an authorization to ensure that funds are available for one or more products that are available for immediate shipment. The authorization request requires no additional fields or requirements other than a basic authorization.

Step 2: Requesting a capture

Ship the available product and request a capture while you wait for the remaining product to become available. The capture request requires only the basic fields as any capture request.

Step 3: Requesting a follow-on capture

When the remaining product becomes available, ship it and request a capture for that amount. The capture request requires only the basic fields as any capture request.

Step 4: Cybersource performs a system-generated authorization

Cybersource performs a system-generated authorization for the follow-on capture request and link it to the original authorization in the Business Center and in reports. Cybersource processes the capture request as a split shipment request because your account is already enabled for split shipments.

Step 5: Cybersource attempts to link the capture request to the other transactions

The capture is linked to the authorizations in the Business Center and in reports through the request IDs as with any capture. All four transactions (first authorization, system-generated authorization, first capture, follow-on capture) are linked together in the Business Center and in reports.

Step 6: Cybersource provides the status

The status of the follow-on capture request and its associated system-generated authorization becomes available.

Related Information

- See *Basic Authorizations* on page 58 for information on how to process a basic authorization.
- See *Captures* on page 112 for information on how to process a capture.

Processing Payments Using Credentials

This section provides the information you need in order to process payments using credentials.

Additional Resources for Credentialed Transactions

For more information, see these guides:

- *Credentialed Transactions Developer Guide*
- *Token Management Service Developer Guide*
- *API field reference guide for the REST API*
- Github repositories: *https://github.com/Cybersource*

Customer-Initiated Transactions with Credentials on File

A customer-initiated transaction (CIT) is a transaction initiated by the customer. There are two types of CITs:

- Customer transactions during which the credentials are stored for future customer-initiated transactions.
- Customer transactions during which the credentials are stored for future merchant-initiated transactions.

Customers can initiate a CIT at a merchant payment terminal, through an online purchase transaction, or by making a purchase using a previously stored credential. When storing cardholder data for a CIT, you must also include 3-D Secure authentication credentials to ensure that the CIT can successfully process. Authentication credentials can be stored for future use with the card credentials by doing a non-payment authentication (NPA).

Business Center

You can create a new customer-initiated transaction in the Business Center by going to the One-Time Payments section and requesting a new authorization. When you have entered the customer's information, you can store the customer's credentials with the customer's permission in the Payment Information section. By doing so, you can perform merchant-initiated transactions for payments that the customer has pre-approved. For more information on how to perform a MIT in the Business Center, see *Merchant-Initiated No-Show Transactions with PAN* on page 346.

Storing Customer Credentials with a CIT and PAN

Before you can perform a merchant-initiated transaction (MIT) or a customer-initiated transaction (CIT) with credentials-on-file (COF), you must store the customer's credentials for later use. Further, before you can store the user's credentials, you must get the customer's consent to store their private information. This is also known as establishing a relationship with the customer.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Storing Customer Credentials During a CIT

Use these required fields for storing customer credentials during a customer-initiated transaction.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator. credentialStoredOnFile	**Set the value to `true`.**

REST Example: Storing Customer Credentials During a CIT

Request

```
{
    "processingInformation": {
        "authorizationOptions": {
            "initiator": {
                "credentialStoredOnFile": "true"
            }
        }
    },
    "orderInformation": {
        "billTo": {
            "firstName": "John",
            "lastName": "Doe",
            "address1": "201 S. Division St.",
            "postalCode": "48104-2201",
            "locality": "Ann Arbor",
            "administrativeArea": "MI",
            "country": "US",
            "email": "test@cybs.com",
            "phoneNumber": "5554327113"
        },
        "amountDetails": {
            "totalAmount": "100.00",
            "currency": "USD"
        }
    },
    "paymentInformation": {
        "card": {
            "expirationYear": "2031",
            "number": "4111xxxxxxxxxxxx",
            "expirationMonth": "12"
        }
    }
}
```

Response to a Successful Request

```
{
    "_links": {
```

```
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6528187198946076303004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1652818719876"
  },
  "id": "6528187198946076303004",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "63165088Z3AHV91G",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-17T20:18:40Z"
}
```

Storing Customer Credentials with a CIT and TMS

Before you can perform a merchant-initiated transaction (MIT) or a customer-initiated transaction (CIT) with credentials-on-file (COF), you must get the customer's consent to store their payment credentials. This is also known as establishing a relationship with the customer. After you have their consent, you can store their payment credentials for later use.

Creating a TMS Token

When sending the initial CIT, you can create a TMS token to store the customer's credentials for the subsequent MITs. To create a TMS token, include the **processingInformation.actionTokenTypes** field in the authorization request. Set the field to one of these values based on the TMS token type you want to create:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "customer"
]
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token. Including a payment instrument in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "paymentInstrument"
]
```

For more information about this TMS token type, see *Payment Instrument Token* in

the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store a PAN. Including an instrument identifier in subsequent MITs eliminates the need to include a PAN and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
    "instrumentIdentifier"
]
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Instrument Identifier, Payment Instrument, and Customer Identifier

You can also create multiple TMS token types in the same authorization. This example includes an instrument identifier, a payment instrument, and a customer token in the same authorization:

```
 "processingInformation": {
   "actionTokenTypes": [
      "instrumentIdentifier",
      "paymentInstrument",
      "customer"
]
```

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Storing Customer Credentials with a CIT and TMS

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.actionList	Set the value to`TOKEN_CREATE`
processingInformation.actionTokenTypes	Set to one or more of these values: - `customer` - `instrumentIdentifier` - `paymnentInstrument`

REST Example: Storing Customer Credentials with a CIT and TMS

Request

```
{
 "processingInformation": {
  "actionList": [
   "TOKEN_CREATE"
  ],
  "actionTokenTypes": [
   "instrumentIdentifier"
  ]
 },
 "paymentInformation": {
  "card": {
   "number": "4111111111111111",
   "expirationMonth": "12",
   "expirationYear": "2031",
   "securityCode": "123"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "4158880000"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6972267090226779103955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6972267090226779103955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6972267090226779103955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "id": "6972267090226779103955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62506622XNMR6Q1Y",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-13T19:51:49Z",
 "tokenInformation": {
  "instrumentidentifierNew": false,
  "instrumentIdentifier": {
   "state": "ACTIVE",
   "id": "7010000000016241111"
  }
 }
}
```

Retrieving Stored Customer Credentials During a CIT

After customers store their credentials on file, you can retrieve these credentials to use with subsequent transactions.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Retrieving Customer Credentials During a Customer-Initiated Transaction

Use these required fields to retrieve customer credentials during a customer-initiated transaction.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator. storedCredentialUsed	**Set field to `true`.**

Card-Specific Required Field for Retrieving Customer Credentials During a CIT

Discover

Discover requires the authorization amount from the original transaction in addition to the above required fields.

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.originalAuthorizedAmount

REST Example: Retrieving Customer Credentials During a CIT

Request

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
        "storedCredentialUsed": "true"
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": "5554327113"
    },
     "amountDetails": {
        "totalAmount": "100.00",
        "currency": "USD",
        "originalAmount": "100"
          // Discover card Only
     }
  },
  "paymentInformation": {
     "card": {
        "expirationYear": "2031",
        "number": "4111xxxxxxxxxxxx",
        "expirationMonth": "12"
     }
  },
 "processorInformation": {
   "transactionId": "12345678961000"
 }
}
```

Response to a Successful Request

```
},
  "paymentAccountInformation": {
     "card": {
        "type": "002"
     }
  },
  "paymentInformation": {
     "tokenizedCard": {
        "type": "002"
     },
     "card": {
        "type": "002"
     }
  },
  "pointOfSaleInformation": {
     "terminalId": "111111"
  },
  "processorInformation": {
     "approvalCode": "888888",
     "authIndicator": "1",
     "networkTransactionId": "123456789619999",
     "transactionId": "123456789619999",
     "responseCode": "100",
     "avs": {
        "code": "X",
        "codeRaw": "I1"
     }
  },
  "reconciliationId": "63740353A3AJ2NSH",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T19:13:06Z"
}
```

Delayed Transaction

Delayed charge transaction is performed to process a supplemental account charge after original services have been rendered and respective payment has been processed.
This section describes how to process a merchant-initiated delayed transaction, also known as a delayed charge, using these payment types:

- *Merchant-Initiated Delayed Transaction with PAN* on page 321
- *Merchant-Initiated Delayed Transaction with TMS* on page 324

Merchant-Initiated Delayed Transaction with PAN

Delayed charge transaction is performed to process a supplemental account charge after original services have been rendered and respective payment has been processed.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing a Merchant-Initiated Delayed Transaction

Use these required fields to process a merchant-initiated delayed transaction.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processorInformation.cardReferenceData	**Required only for token transactions with Discover or Diners Club. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.**
processingInformation. authorizationOptions.initiator. merchantInitiatedTransaction. previousTransactionId	- American Express: set to the transaction ID from the original transaction. - Discover: set to the transaction ID from the original transaction. - Visa: set to the last successful transaction ID.
processingInformation.authorizationOptions.initiator. merchantInitiatedTransaction.reason	**Set the value to `2`.** **Required only for Discover, Mastercard, and Visa.**
processingInformation. authorizationOptions. initiator. type	**Set the value to `merchant`.**
issuerInformation.transactionInformation	**Required only for token transactions with Discover or Diners Club. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**

Card-Specific Required Field for Processing a Merchant-Initiated Transactions

Discover

The listed card requires an additional field:

processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. originalAuthorizedAmount	**Provide the original transaction amount.**

REST Example: Processing a Merchant-Initiated Delayed Authorization Transaction

Request

```
{
  "orderInformation": {
 "billTo" : {
   "country" : "US",
   "lastName" : "Kim",
   "address1" : "201 S. Division St.",
   "postalCode" : "48104-2201",
   "locality" : "Ann Arbor",
   "administrativeArea" : "MI",
   "firstName" : "Kyong-Jin",
      "phoneNumber": "5554327113",
   "email" : "test@cybs.com"
   },
    "amountDetails": {
      "totalAmount": "120.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  },
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
     "type": "merchant",
       "merchantInitiatedTransaction": {
       "originalAuthorizedAmount": "100",
         // Discover only
       "previousTransactionId": "123456789619999",
       "reason": "2"
       }
      }
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6534213653516599003001/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6534213653516599003001"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6534213653516599003001/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653421365327"
  },
  "id": "6534213653516599003001",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "120.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "64365475T3K10Q1D",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-24T19:42:45Z"
}
```

Merchant-Initiated Delayed Transaction with TMS

Delayed charge transaction is performed to process a supplemental account charge after original services have been rendered and respective payment has been processed.
This section describes how to process a merchant-initiated delayed transaction using these TMS token types:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
 "customer": {
  "id": "07C9CA98022DA498E063A2598D0AA400"
 }
}
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token.

Including a payment instrument eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
 "paymentInstrument": {
  "id": "07CA24EF20F9E2C9E063A2598D0A8565"
 }
}
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store only a PAN. Including an instrument identifier eliminates the need to include a PAN and the previous transaction's ID.

```
"paymentInformation": {
 "instrumentIdentifier": {
  "id": "7010000000016241111"
 }
}
```

	For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for MIT Delayed Transaction with TMS

Include these Required Fields

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.[tokentype].id	Where **[tokentype]** is the TMS token type you are using: • **customer** • **instrumentIdentifier** • **paymentInstrument**
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason	Set the value to `2`. Required only for Discover, Mastercard, and Visa.

Instrument Identifier Required Fields

If you are using the **paymentInformation.instrumentIdentifier.id** token, include these required fields in addition to the required fields listed above.

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

Card-Specific Fields

Include these fields when processing an authorization with these card types. The listed card type requires an additional field.

Diners Club	**processorInformation.cardReferenceData:** **Required only for token transactions. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.** **issuerInformation.transactionInformation:** **Required only for token transactions. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**
Discover	**processingInformation.authorizationOptions.initiator. merchantInitiatedTransaction.originalAuthorizedAmou** **Set to the original transaction amount.** **processorInformation.cardReferenceData** **Required only for token transactions. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.** **issuerInformation.transactionInformation** **Required only for token transactions. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**

Example: MIT Delayed Transaction with TMS Instrument Identifier

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "2"
    }
   }
  }
 },
 "paymentInformation": {
  "card": {
   "expirationMonth": "12",
   "expirationYear": "2031"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "4158880000"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976922830456934003954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976922830456934003954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976922830456934003954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697692283160"
 },
 "id": "6976922830456934003954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62700184NNMR6XFK",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T05:11:23Z"
}
```

Example: MIT Delayed Transaction with TMS Payment Instrument

Request

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
        "merchantInitiatedTransaction": {
          "reason": "2"
        }
      }
    }
  },
  "paymentInformation": {
    "paymentInstrument": {
      "id": "080AE120369A7947E063A2598D0A718F"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "USD"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976917718796256603955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6976917718796256603955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976917718796256603955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697691771976"
  },
  "id": "6976917718796256603955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
```

```
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62700629BNN13VGW",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T05:02:52Z"
}
```

Example: MIT Delayed Transaction with TMS Customer token

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "2"
    }
   }
  }
 },
 "paymentInformation": {
  "customer": {
   "id": "080AC9AB60C92AA2E063A2598D0A0C74"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
 {
  "_links": {
   "authReversal": {
    "method": "POST",
    "href": "/pts/v2/payments/6976916433716228003955/reversals"
   },
   "self": {
    "method": "GET",
    "href": "/pts/v2/payments/6976916433716228003955"
   },
   "capture": {
    "method": "POST",
    "href": "/pts/v2/payments/6976916433716228003955/captures"
   }
  },
  "clientReferenceInformation": {
   "code": "1697691643458"
  },
  "id": "6976916433716228003955",
  "orderInformation": {
   "amountDetails": {
    "authorizedAmount": "102.21",
    "currency": "USD"
   }
  },
  "paymentAccountInformation": {
   "card": {
    "type": "001"
   }
  },
  "paymentInformation": {
   "tokenizedCard": {
    "type": "001"
   },
   "instrumentIdentifier": {
    "id": "7010000000016241111",
    "state": "ACTIVE"
   },
   "paymentInstrument": {
    "id": "080AE6DB37B09557E063A2598D0AA4C9"
   },
   "card": {
    "type": "001"
   },
   "customer": {
    "id": "080AC9AB60C92AA2E063A2598D0A0C74"
   }
  },
  "pointOfSaleInformation": {
   "terminalId": "111111"
  },
  "processingInformation": {
   "paymentSolution": "015"
  },
  "processorInformation": {
   "paymentAccountReferenceNumber": "V0010013022298169667504231315",
   "approvalCode": "888888",
   "networkTransactionId": "123456789619999",
   "transactionId": "123456789619999",
   "responseCode": "100",
   "avs": {
    "code": "X",
    "codeRaw": "I1"
   }
  },
  "reconciliationId": "62700435FNN143RY",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:00:43Z"
 }
```

Incremental Transaction

An incremental authorization is used to increase the total amount authorized for a payment if the initial authorization does not cover the total cost of goods and services. An incremental transaction is an additional amount to the original authorization. The final authorized total includes amounts for both the initial and the incremental authorizations. Incremental transactions are limited to certain merchant categories, such as rental, lodging, transit, amusement parks, restaurants, and bars.

This section describes how to process an incremental transaction using these payment types:

- *Payment Account Number (PAN)*
- *Token Management Service (TMS)*

Merchant-Initiated Incremental Transaction with PAN

An incremental authorization is used to increase the total amount authorized for a payment if the initial authorization does not cover the total cost of goods and services. An incremental transaction is an additional amount to the original authorization. The final authorized total includes amounts for both the initial and the incremental authorizations. Incremental transactions are limited to certain merchant categories, such as rental, lodging, transit, amusement parks, restaurants, and bars.

To create an incremental transaction using the Business Center, choose one of these options:

- Account Top Up
- No Show

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing Merchant-Initiated Incremental Transactions

Use these required fields to process merchant-initiated incremental transactions.

issuerInformation.transactionInformation	**Required only for token transactions with Discover or Diners Club. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. previousTransactionId	
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason	**Set the value to `5`.** **Required only for Discover and Visa.**
processingInformation. authorizationOptions.initiator. type	**Set the value to `merchant`.**
processorInformation.cardReferenceData	**Required only for token transactions with Discover or Diners Club. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.**

Card-Specific Required Field for Processing a Merchant-Initiated Transactions

Discover

The listed card requires an additional field:

processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. originalAuthorizedAmount	**Provide the original transaction amount.**

REST Example: Processing Merchant-Initiated Incremental Transactions

Request

```
{
   "orderInformation": {
  "billTo" : {
     "country" : "US",
     "lastName" : "Kim",
     "address1" : "201 S. Division St.",
     "postalCode" : "48104-2201",
     "locality" : "Ann Arbor",
     "administrativeArea" : "MI",
     "firstName" : "Kyong-Jin",
         "phoneNumber": "5554327113",
     "email" : "test@cybs.com"
    },
      "amountDetails": {
         "totalAmount": "120.00",
         "currency": "USD"
      }
   },
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "number": "4111xxxxxxxxxxxx",
         "expirationMonth": "12"
      }
   },
   "processingInformation": {
      "authorizationOptions": {
         "initiator": {
       "type": "merchant",
          "merchantInitiatedTransaction": {
           "originalAuthorizedAmount": "100",
                    // Required for Discover
           "previousTransactionId": "123456789619999",
           "reason": "5"
          }
         }
      }
   }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6533225006556860003002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6533225006556860003002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6533225006556860003002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653322500637"
  },
  "id": "6533225006556860003002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "64143477A3AJ4P2Z",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-23T16:15:00Z"
}
```

Merchant-Initiated Incremental Transaction with TMS

An incremental authorization is used to increase the total amount authorized for a payment if the initial authorization does not cover the total cost of goods and services. An incremental transaction is an additional amount to the original authorization. The final authorized total includes amounts for both the initial and the incremental authorizations. Incremental transactions are limited to certain merchant categories, such as rental, lodging, transit, amusement parks, restaurants, and bars.

This section describes how to process a merchant-initiated incremental transaction using these TMS token types:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "customer": {
    "id": "07C9CA98022DA498E063A2598D0AA400"
  }
}
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token.

Including a payment instrument eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "paymentInstrument": {
    "id": "07CA24EF20F9E2C9E063A2598D0A8565"
  }
}
```

For more information about this TMS token type, see *Payment Instrument Token* in

	the Token Management Service Developer Guide.

Instrument Identifier	Instrument identifier tokens store only a PAN. Including an instrument identifier eliminates the need to include a PAN and the previous transaction's ID.
	`"paymentInformation": {`
	`  "instrumentIdentifier": {`
	`    "id": "7010000000016241111"`
	`  }`
	`}`
	For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

To create an incremental transaction using the Business Center, choose one of these options:

- Account Top Up
- No Show

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for MIT Incremental Transaction with TMS

Include these Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.[tokentype].id	Where **[tokentype]** is the TMS token type you are using: • **customer** • **instrumentIdentifier** • **paymentInstrument**

processingInformation. authorizationOptions.initiator. merchantInitiatedTransaction. reason	Set the value to 5. Required only for Discover and Visa.

Instrument Identifier Required Fields

If you are using the **paymentInformation.instrumentIdentifier.id** token, include these required fields in addition to the required fields listed above.

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

Card-Specific Fields

Include these fields when processing an authorization with these card types.
The listed card type requires an additional field.

Diners Club	**processorInformation.cardReferenceData:** **Required only for token transactions. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.** **issuerInformation.transactionInformation:** **Required only for token transactions. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**
Discover	**processingInformation.authorizationOptions.initiator. merchantInitiatedTransaction.originalAuthorizedAmou** **Set to the original transaction amount.**

processorInformation.cardReferenceData

Required only for token transactions. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.

issuerInformation.transactionInformation

Required only for token transactions. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.

Example: MIT Incremental Transaction with a TMS Instrument Identifier

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "5"
    }
   }
  }
 },
 "paymentInformation": {
  "card": {
   "expirationMonth": "12",
   "expirationYear": "2031"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "4158880000"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976922830456934003954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976922830456934003954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976922830456934003954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697692283160"
 },
 "id": "6976922830456934003954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62700184NNMR6XFK",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T05:11:23Z"
}
```

Example: MIT Incremental Transaction with a TMS Payment Instrument

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "5"
    }
   }
  }
 },
 "paymentInformation": {
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976917718796256603955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697691771976"
 },
 "id": "6976917718796256603955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62700629BNN13VGW",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T05:02:52Z"
}
```

Example: MIT Incremental Transaction with a TMS Customer token

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "5"
    }
   }
  }
 },
 "paymentInformation": {
  "customer": {
   "id": "080AC9AB60C92AA2E063A2598D0A0C74"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976916433716228003955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976916433716228003955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976916433716228003955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697691643458"
 },
 "id": "6976916433716228003955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "080AE6DB37B09557E063A2598D0AA4C9"
  },
  "card": {
   "type": "001"
  },
  "customer": {
   "id": "080AC9AB60C92AA2E063A2598D0A0C74"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62700435FNN143RY",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T05:00:43Z"
}
```

No-Show Transactions

A no-show authorization occurs when a merchant charges a customer after the customer makes a reservation, and does not show up to claim the reservation. In this situation, the customer is charged an agreed upon fee for not showing up as expected.

This section describes how to process a merchant-initiated no-show transaction using these payment types:

- *Merchant-Initiated No-Show Transactions with PAN* on page 346

- *Merchant-Initiated No-Show Transaction with TMS* on page 349

Merchant-Initiated No-Show Transactions with PAN

A no-show authorization occurs when a merchant charges a customer after the customer makes a reservation, and does not show up to claim the reservation. In this situation, the customer is charged an agreed upon fee for not showing up as expected.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing Merchant-Initiated No-Show Charges

Use these required fields to process a merchant-initiated no-show charges transaction.

issuerInformation.transactionInformation	**Required only for token transactions with Discover or Diners Club. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	

paymentInformation.card.number	
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. previousTransactionId	• American Express: set to the transaction ID from the original transaction. • Discover: set to the transaction ID from the original transaction. • Visa: set to the last successful transaction ID.
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason	**Set the value to** 4**.** **Required only for Discover, Mastercard, and Visa.**
processingInformation. authorizationOptions. initiator. type	**Set the value to** merchant**.**
processorInformation.cardReferenceData	**Required only for token transactions with Discover or Diners Club. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.**

Card-Specific Required Field for Processing a Merchant-Initiated Transactions

Discover

The listed card requires an additional field:

processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. originalAuthorizedAmount	**Provide the original transaction amount.**

Optional Field for Processing Merchant-Initiated No-Show Charges

You can use these optional fields to include additional information when authorizing a request for an MIT no-show charge:

processingInformation. authorizationOptions. initiator. storedCredentialUsed	**If the payment information is COF information, set to** true**.**

REST Example: Processing Merchant-Initiated No-Show Transactions

Request

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
     "type": "merchant",
        "merchantInitiatedTransaction": {
         "originalAuthorizedAmount": "100", //Discover only
         "previousTransactionId": "123456789619999",
         "reason": "4"
        }
       }
     }
   },
   "orderInformation": {
  "billTo" : {
    "country" : "US",
    "lastName" : "Kim",
    "address1" : "201 S. Division St.",
    "postalCode" : "48104-2201",
    "locality" : "Ann Arbor",
    "administrativeArea" : "MI",
    "firstName" : "Kyong-Jin",
       "phoneNumber": "5554327113",
    "email" : "test@cybs.com"
   },
     "amountDetails": {
       "totalAmount": "150.00",
       "currency": "USD"
     }
   },
   "paymentInformation": {
     "card": {
       "expirationYear": "2031",
       "number": "4111xxxxxxxxxxxx",
       "expirationMonth": "12"
     }
   }
}
```

Response to a Successful Request

```
{
   "_links": {
     "authReversal": {
       "method": "POST",
       "href": "/pts/v2/payments/6534214295466223903006/reversals"
     },
     "self": {
       "method": "GET",
       "href": "/pts/v2/payments/6534214295466223903006"
     },
     "capture": {
       "method": "POST",
       "href": "/pts/v2/payments/6534214295466223903006/captures"
     }
   },
   "clientReferenceInformation": {
     "code": "1653421429522"
   },
   "id": "6534214295466223903006",
   "orderInformation": {
     "amountDetails": {
       "authorizedAmount": "150.00",
       "currency": "USD"
     }
   },
   "paymentAccountInformation": {
     "card": {
       "type": "001"
     }
   },
   "paymentInformation": {
     "tokenizedCard": {
       "type": "001"
     },
     "card": {
       "type": "001"
     }
   },
   "pointOfSaleInformation": {
     "terminalId": "111111"
   },
   "processorInformation": {
     "approvalCode": "888888",
     "networkTransactionId": "123456789619999",
     "transactionId": "123456789619999",
     "responseCode": "100",
     "avs": {
       "code": "X",
       "codeRaw": "I1"
     }
   },
   "reconciliationId": "64365823G3K7HFAM",
   "status": "AUTHORIZED",
   "submitTimeUtc": "2022-05-24T19:43:49Z"
}
```

Merchant-Initiated No-Show Transaction with TMS

A no-show authorization occurs when a merchant charges a customer after the customer makes a reservation, and does not show up to claim the reservation. In this situation, the customer is charged an agreed upon fee for not showing up as expected.

This section describes how to process a merchant-initiated no-show transaction using these TMS token types:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
 "customer": {
  "id": "07C9CA98022DA498E063A2598D0AA400"
 }
}
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token.

Including a payment instrument eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
 "paymentInstrument": {
  "id": "07CA24EF20F9E2C9E063A2598D0A8565"
 }
}
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store only a PAN. Including an instrument identifier eliminates the need to include a PAN and the previous transaction's ID.

```
"paymentInformation": {
 "instrumentIdentifier": {
  "id": "7010000000016241111"
 }
}
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard

- Visa

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for MIT No-Show Transaction with TMS

Include these Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.[tokentype].id	Where **[tokentype]** is the TMS token type you are using: • customer • instrumentIdentifier • paymentInstrument
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction.reason	Set the value to 4. Required only for Discover, Mastercard, and Visa.

Instrument Identifier Required Fields

If you are using the **paymentInformation.instrumentIdentifier.id** token, include these required fields in addition to the required fields listed above.

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

Card-Specific Fields

Include these fields when processing an authorization with these card types.

The listed card type requires an additional field.

Diners Club	**processorInformation.cardReferenceData:** **Required only for token transactions. Set this field to the processorInformation.cardReferenceDatafield that was in the response message when you obtained the customer's credentials.** **issuerInformation.transactionInformation:** **Required only for token transactions. Set this field to the processorInformation.transactionIDfield that was in the response message when you obtained the customer's credentials.**
Discover	**processingInformation.authorizationOptions.initiator. merchantInitiatedTransaction.originalAuthorizedAmou** **Set to the original transaction amount.** **processorInformation.cardReferenceData** **Required only for token transactions. Set this field to the processorInformation.cardReferenceDatafield that was in the response message when you obtained the customer's credentials.** **issuerInformation.transactionInformation** **Required only for token transactions. Set this field to the processorInformation.transactionIDfield that was in the response message when you obtained the customer's credentials.**

Example: MIT No-Show Transaction with a TMS Instrument Identifier

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "4"
    }
   }
  }
 },
 "paymentInformation": {
  "card": {
   "expirationMonth": "12",
   "expirationYear": "2031"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "4158880000"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976922830456934003954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976922830456934003954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976922830456934003954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697692283160"
 },
 "id": "6976922830456934003954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111",
      "state": "ACTIVE"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processingInformation": {
    "paymentSolution": "015"
  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "V0010013022298169667504231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62700184NNMR6XFK",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:11:23Z"
}
```

Example: MIT No-Show Transaction with a TMS Payment Instrument

Request

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
        "merchantInitiatedTransaction": {
          "reason": "4"
        }
      }
    }
  },
  "paymentInformation": {
    "paymentInstrument": {
      "id": "080AE120369A7947E063A2598D0A718F"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976917718796256603955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697691771976"
 },
 "id": "6976917718796256603955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
    "paymentSolution": "015"
  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "V0010013022298169667504231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62700629BNN13VGW",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:02:52Z"
}
```

Example: MIT No-Show Transaction with a TMS Customer

Request

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
        "merchantInitiatedTransaction": {
          "reason": "4"
        }
      }
    }
  },
  "paymentInformation": {
    "customer": {
      "id": "080AC9AB60C92AA2E063A2598D0A0C74"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "USD"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6976916433716228003955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697691643458"
  },
  "id": "6976916433716228003955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111",
      "state": "ACTIVE"
    },
    "paymentInstrument": {
      "id": "080AE6DB37B09557E063A2598D0AA4C9"
    },
    "card": {
      "type": "001"
    },
    "customer": {
      "id": "080AC9AB60C92AA2E063A2598D0A0C74"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processingInformation": {
    "paymentSolution": "015"
  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "V0010013022298169667504231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62700435FNN143RY",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:00:43Z"
}
```

Reauthorization Transaction

A reauthorization occurs when the completion or fulfillment of the original order or service extends beyond the authorized amount time limit. There are two common reauthorization scenarios:

- Split or delayed shipments by a retailer
- Extended car rentals, hotel stays, or cruise line bookings

This section describes how to process a reauthorization transaction using these payment methods:

- *Merchant-Initiated Reauthorization Transactions with PAN* on page 358
- *Merchant-Initiated Reauthorization Transactions with TMS* on page 361

Merchant-Initiated Reauthorization Transactions with PAN

A reauthorization occurs when the completion or fulfillment of the original order or service extends beyond the authorized amount time limit. There are two common reauthorization scenarios:

- Split or delayed shipments by a retailer
- Extended car rentals, hotel stays, or cruise line bookings

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for Processing Merchant-Initiated Reauthorized Transactions

Use these required fields to process a merchant-initiated reauthorization transaction.

Important

> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. previousTransactionId	• American Express: set to the transaction ID from the original transaction. • Discover: set to the transaction ID from the original transaction. • Visa: set to the last successful transaction ID.
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason	**Set the value to `3`.** **Required only for Discover and Visa.**
processingInformation. authorizationOptions. initiator. type	**Set the value to `merchant`.**

Card-Specific Required Field for Processing a Merchant-Initiated Transactions

Discover

The listed card requires an additional field:

processingInformation. authorizationOptions. initiator.	**Provide the original transaction amount.**

merchantInitiatedTransaction. originalAuthorizedAmount

REST Example: Processing a Merchant-Initiated Reauthorized Transaction

Request

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
     "type": "merchant",
       "merchantInitiatedTransaction": {
        "originalAuthorizedAmount": "100", // Discover Only
        "previousTransactionId": "123456789619999",
        "reason": "3"
       }
      }
    }
  },
  "orderInformation": {
 "billTo" : {
   "country" : "US",
   "lastName" : "Kim",
   "address1" : "201 S. Division St.",
   "postalCode" : "48104-2201",
   "locality" : "Ann Arbor",
   "administrativeArea" : "MI",
   "firstName" : "Kyong-Jin",
      "phoneNumber": "5554327113",
   "email" : "test@cybs.com"
  },
    "amountDetails": {
      "totalAmount": "130.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6541178668686490403003/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6541178668686490403003"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6541178668686490403003/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1654117866849"
  },
  "id": "6541178668686490403003",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "130.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "65313868D3TXXC05",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-06-01T21:11:06Z"
}
```

Merchant-Initiated Reauthorization Transactions with TMS

A reauthorization occurs when the completion or fulfillment of the original order or service extends beyond the authorized amount time limit. There are two common reauthorization scenarios:

- Split or delayed shipments by a retailer

- Extended car rentals, hotel stays, or cruise line bookings

This section describes how to process a merchant-initiated reauthorization transactions using one or more TMS token types:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "customer": {
    "id": "07C9CA98022DA498E063A2598D0AA400"
  }
}
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token.

Including a payment instrument eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "paymentInstrument": {
    "id": "07CA24EF20F9E2C9E063A2598D0A8565"
  }
}
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store only a PAN. Including an instrument identifier eliminates the need to include a PAN and the previous transaction's ID.

```
"paymentInformation": {
  "instrumentIdentifier": {
   "id": "7010000000016241111"
  }
 }
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for MIT Reauthorization Transaction with TMS

Include these Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.[tokentype].id	Where **[tokentype]** is the TMS token type you are using: • **customer** • **instrumentIdentifier** • **paymentInstrument**
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason	Set the value to `3`. Required only for Discover and Visa.

Instrument Identifier Required Fields

If you are using the **paymentInformation.instrumentIdentifier.id** token, include these required fields in addition to the required fields listed above.

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

Card-Specific Fields

Include these fields when processing an authorization with these card types. The listed card type requires an additional field.

Diners Club	**processorInformation.cardReferenceData:** **Required only for token transactions. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.** **issuerInformation.transactionInformation:** **Required only for token transactions. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**
Discover	**processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.originalAuthorizedAmou** **Set to the original transaction amount.** **processorInformation.cardReferenceData** **Required only for token transactions. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.** **issuerInformation.transactionInformation** **Required only for token transactions. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**

Example: MIT Reauthorization Transaction with a TMS Instrument Identifier

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "3"
    }
   }
  }
 },
 "paymentInformation": {
  "card": {
   "expirationMonth": "12",
   "expirationYear": "2031"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "4158880000"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976922830456934003954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976922830456934003954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976922830456934003954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697692283160"
 },
 "id": "6976922830456934003954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62700184NNMR6XFK",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T05:11:23Z"
}
```

Example: MIT Reauthorization Transaction with a TMS Payment Instrument

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "3"
    }
   }
  }
 },
 "paymentInformation": {
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976917718796256603955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697691771976"
 },
 "id": "6976917718796256603955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62700629BNN13VGW",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T05:02:52Z"
}
```

Example: MIT Reauthorization Transaction with a TMS Customer

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "3"
    }
   }
  }
 },
 "paymentInformation": {
  "customer": {
      "id": "080AC9AB60C92AA2E063A2598D0A0C74"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "USD"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6976916433716228003955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697691643458"
  },
  "id": "6976916433716228003955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111",
      "state": "ACTIVE"
    },
    "paymentInstrument": {
      "id": "080AE6DB37B09557E063A2598D0AA4C9"
    },
    "card": {
      "type": "001"
```

```
   },
   "customer": {
    "id": "080AC9AB60C92AA2E063A2598D0A0C74"
   }
  },
  "pointOfSaleInformation": {
   "terminalId": "111111"
  },
  "processingInformation": {
   "paymentSolution": "015"
  },
  "processorInformation": {
   "paymentAccountReferenceNumber": "V0010013022298169667504231315",
   "approvalCode": "888888",
   "networkTransactionId": "123456789619999",
   "transactionId": "123456789619999",
   "responseCode": "100",
   "avs": {
    "code": "X",
    "codeRaw": "I1"
   }
  },
  "reconciliationId": "62700435FNN143RY",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:00:43Z"
}
```

Resubmission Transaction

A resubmission transaction is an authorization that you resubmit to recover an outstanding debt from the customer. A common scenario is when a card was initially declined due to insufficient funds, but the goods or services were already delivered to the customer.

You can request the resubmission transaction with a PAN or a TMS token.

Merchant-Initiated Resubmission Transaction with PAN

A resubmission transaction is an authorization that you resubmit to recover an outstanding debt from the customer. A common scenario is when a card was initially declined due to insufficient funds, but the goods or services were already delivered to the customer.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Processing a Merchant-Initiated Resubmitted Transaction

Use these required fields to process a merchant-initiated resubmitted transaction.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation. authorizationOptions. initiator.merchantInitiatedTransaction. previousTransactionId	• American Express: set to the transaction ID from the original transaction. • Discover: set to the transaction ID from the original transaction. • Visa: set to the last successful transaction ID.
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason	**Set the value to `1`.** **Required only for Discover, Mastercard, and Visa.**
processingInformation. authorizationOptions. initiator. type	**Set the value to `merchant`.**

Card-Specific Required Field for Processing a Merchant-Initiated Transactions

Discover

The listed card requires an additional field:

processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. originalAuthorizedAmount	**Provide the original transaction amount.**

REST Example: Processing a Merchant-Initiated Resubmitted Transaction

Request

```
{
   "processingInformation": {
      "authorizationOptions": {
         "initiator": {
       "type": "merchant",
          "merchantInitiatedTransaction": {
           "originalAuthorizedAmount": "100",  // Discover Only
           "previousTransactionId": "123456789619999",
           "reason": "1"
          }
         }
      }
   },
   "orderInformation": {
 "billTo" : {
    "country" : "US",
    "lastName" : "Kim",
    "address1" : "201 S. Division St.",
    "postalCode" : "48104-2201",
    "locality" : "Ann Arbor",
    "administrativeArea" : "MI",
    "firstName" : "Kyong-Jin",
         "phoneNumber": "5554327113",
    "email" : "test@cybs.com"
  },
      "amountDetails": {
         "totalAmount": "100.00",
         "currency": "USD"
      }
   },
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "number": "4111xxxxxxxxxxxx",
         "expirationMonth": "12"
      }
   }
}
```

Response to a Successful Request

```
{
   "_links": {
      "authReversal": {
         "method": "POST",
         "href": "/pts/v2/payments/6534232293716260503006/reversals"
```

```
            },
            "self": {
                "method": "GET",
                "href": "/pts/v2/payments/6534232293716260503006"
            },
            "capture": {
                "method": "POST",
                "href": "/pts/v2/payments/6534232293716260503006/captures"
            }
        },
        "clientReferenceInformation": {
            "code": "1653423229353"
        },
        "id": "6534232293716260503006",
        "orderInformation": {
            "amountDetails": {
                "authorizedAmount": "100.00",
                "currency": "USD"
            }
        },
        "paymentAccountInformation": {
            "card": {
                "type": "004"
            }
        },
        "paymentInformation": {
            "tokenizedCard": {
                "type": "004"
            },
            "card": {
                "type": "004"
            }
        },
        "pointOfSaleInformation": {
            "terminalId": "111111"
        },
        "processorInformation": {
            "approvalCode": "888888",
            "networkTransactionId": "123456789619999",
            "transactionId": "123456789619999",
            "responseCode": "100",
            "avs": {
                "code": "X",
                "codeRaw": "I1"
            }
        },
        "reconciliationId": "64365912G3K7HFDJ",
        "status": "AUTHORIZED",
        "submitTimeUtc": "2022-05-24T20:13:49Z"
    }
```

Merchant-Initiated Resubmission Transaction with TMS

A resubmission transaction is an authorization that you resubmit to recover an outstanding debt from the customer. A common scenario is when a card was initially declined due to insufficient funds, but the goods or services were already delivered to the customer.

This section describes how to process a merchant-initiated resubmission transaction using these TMS token types:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "customer": {
    "id": "07C9CA98022DA498E063A2598D0AA400"
  }
}
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token.

Including a payment instrument eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "paymentInstrument": {
    "id": "07CA24EF20F9E2C9E063A2598D0A8565"
  }
}
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store only a PAN. Including an instrument identifier eliminates the need to include a PAN and the previous transaction's ID.

```
"paymentInformation": {
  "instrumentIdentifier": {
    "id": "70100000000016241111"
  }
}
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for MIT Resubmission Transaction with TMS

Include these Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.[tokentype].id	Where **[tokentype]** is the TMS token type you are using: • **customer** • **instrumentIdentifier** • **paymentInstrument**
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason	Set the value to 1. Required only for Discover, Mastercard, and Visa.

Instrument Identifier Required Fields

If you are using the **paymentInformation.instrumentIdentifier.id** token, include these required fields in addition to the required fields listed above.

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

Card-Specific Fields

Include these fields when processing an authorization with these card types. The listed card type requires an additional field.

Diners Club	**processorInformation.cardReferenceData:** **Required only for token transactions. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.** **issuerInformation.transactionInformation:** **Required only for token transactions. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**
Discover	**processingInformation.authorizationOptions.initiator. merchantInitiatedTransaction.originalAuthorizedAmou** **Set to the original transaction amount.** **processorInformation.cardReferenceData** **Required only for token transactions. Set this field to the processorInformation.cardReferenceData field that was in the response message when you obtained the customer's credentials.** **issuerInformation.transactionInformation** **Required only for token transactions. Set this field to the processorInformation.transactionID field that was in the response message when you obtained the customer's credentials.**

Example: MIT Resubmission Transaction with a TMS Instrument Identifier

Request

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
        "merchantInitiatedTransaction": {
          "reason": "1"
        }
      }
    }
  },
  "paymentInformation": {
    "card": {
      "expirationMonth": "12",
      "expirationYear": "2031"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "USD"
    },
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "1 Market St",
      "locality": "san francisco",
      "administrativeArea": "CA",
      "postalCode": "94105",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": "4158880000"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976922830456934003954/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6976922830456934003954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976922830456934003954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697692283160"
  },
  "id": "6976922830456934003954",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111",
      "state": "ACTIVE"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processingInformation": {
    "paymentSolution": "015"
  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "V0010013022298169667504231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62700184NNMR6XFK",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:11:23Z"
}
```

Example: MIT Resubmission Transaction with a TMS Payment Instrument

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "1"
    }
   }
  }
 },
 "paymentInformation": {
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976917718796256603955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697691771976"
 },
 "id": "6976917718796256603955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62700629BNN13VGW",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T05:02:52Z"
}
```

Example: MIT Reauthorization Transaction with a TMS Customer

Request

```
{
 "processingInformation": {
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "1"
    }
   }
  }
 },
 "paymentInformation": {
  "customer": {
   "id": "080AC9AB60C92AA2E063A2598D0A0C74"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976916433716228003955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976916433716228003955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976916433716228003955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697691643458"
 },
 "id": "6976916433716228003955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "080AE6DB37B09557E063A2598D0AA4C9"
  },
  "card": {
   "type": "001"
  },
  "customer": {
   "id": "080AC9AB60C92AA2E063A2598D0A0C74"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62700435FNN143RY",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T05:00:43Z"
}
```

Installment Payments

An installment payment is a single purchase of goods or services billed to a customer in multiple transactions over a period of time agreed to by you and the customer. The agreement enables you to charge a specific amount at specified intervals.

Installments Service for Installment Payments

> **Important**
>
> Do not use this document if you are using the Installments service. When using the Installments service, Cybersource saves and stores payment credentials for installment transactions, ensuring compliance with COF best practices.

Customer-Initiated Installment Payments with PAN

An installment payment is a single purchase of goods or services billed to a customer in multiple transactions over a period of time agreed to by you and the customer, and sometimes, the issuing bank. The agreement enables you to charge a specific amount at specified intervals. For customers, installment payments provide greater purchasing power and lower impact on their monthly budget. For you, offering installment payments at checkout can help increase the number of successfully completed purchases.

Before you can accept installment payments, you and your acquirer must agree on the maximum number of installments you can accept, which can be different for each card type.

In Brazil, installment payments are also known as parcelados and parcelas.

> **Important**
>
> Do not use this document if you are using the Installments service. When using the Installments service, Cybersource saves and stores payment credentials for installment transactions, ensuring compliance with COF best practices.

Installment Payment Types

Visa Platform Connect enables you to process installment payments but does not have a role in setting the terms for the installment plan.

Visa Platform Connect enables you to process these types of installments payments:

Issuer-Funded Installment Payments	The customer pays for goods or services using an installment plan agreed upon by the customer and their issuing bank. The issuer controls how the customer's account is debited. Your account is credited for the entire amount in a single transaction. The issuer assumes the risk and establishes credit rates and fees that are charged to the customer. The customer pays the funding cost, which is a fee for paying in installments. In Brazil, a Crediario is a special type of issuer-funded installment payment plan that enables the customer to request information about the terms of the installment plan before approving the installment payments.
Merchant-Funded Installment Payments	The customer pays for goods or services using an installment plan agreed upon by you and the customer. The issuer controls how the customer's account is debited. Your account is credited periodically for partial amounts as the customer's account is debited. You assume the risk and establish the credit rate and fees that are charged to the customer.
Co-Branded Merchant Financed Installment Payments—Brazil Only	You and the issuer determine the terms for this kind of installment plan. The funding varies depending on the agreement between you, the issuer, and the customer.

	This funding method is available only for Mastercard installment payments in Brazil.
Issuer Merchant Co-Financed Installment Payments—Brazil Only	The issuer creates the installment plan. You and the issuer determine the service fees that the customer pays to you and the issuer. The acquirer is paid in full while the issuer is paid in installments by the customer. You or the customer pay the funding cost, which is a fee for paying in installments. This funding method is available only for Mastercard installment payments in Brazil.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Successful Response

You must store the network transaction ID from the successful response message to include in subsequent MIT authorization requests in order to associate the CIT to the MIT. The network transaction ID is the **processorInformation.networkTransactionId** field value. Store the network transaction ID, which is the **processorInformation.networkTransactionId** field value, from the successful response message. You must include the network transaction ID in subsequent MIT authorization requests in order to associate the CIT to the MIT.

Required Fields for Initial Customer-Initiated Installment Payments with a PAN

Include these required fields to authorize an initial customer-initiated installment payment using a PAN.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation. authorizationOptions. initiator. credentialStoredOnFile	Set the value to `true`.
processingInformation. authorizationOptions. initiator. type	Set the value to `customer`.
processingInformation. commerceIndicator	Set the value to `internet`, `MOTO`, or a payer authentication value.

Card-Specific Fields for Authorizing Initial Installment Payments

Use this required field if you are authorizing an initial installment payment using the card type referenced below.

Mastercard	processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason Set the value to `9`.

REST Example: Authorizing Initial Customer-Initiated Installment Payments with a PAN

Request

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "type": "customer",
        "credentialStoredOnFile": "true",
        "merchantInitiatedTransaction": {
          "reason": "9"   //Mastercard only
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "5554327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6528187198946076303004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1652818719876"
  },
  "id": "6528187198946076303004",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "63165088Z3AHV91G",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-17T20:18:40Z"
}
```

Customer-Initiated Installment Payment with TMS

An installment payment is a single purchase of goods or services billed to a customer in multiple transactions over a period of time agreed to by you and the customer, and sometimes, the issuing bank. The agreement enables you to charge a specific amount at specified intervals. For customers, installment payments provide greater purchasing power and lower impact on their monthly budget. For you, offering installment payments at checkout can help increase the number of successfully completed purchases.

Before you can accept installment payments, you and your acquirer must agree on the maximum number of installments you can accept, which can be different for each card type.

In Brazil, installment payments are also known as parcelados and parcelas.

> **Important**
>
> Do not use this document if you are using the Installments service. When using the Installments service, Cybersource saves and stores payment credentials for installment transactions, ensuring compliance with COF best practices.

Installment Payment Types

Visa Platform Connect enables you to process installment payments but does not have a role in setting the terms for the installment plan.

Visa Platform Connect enables you to process these types of installments payments:

Issuer-Funded Installment Payments	The customer pays for goods or services using an installment plan agreed upon by the customer and their issuing bank. The issuer controls how the customer's account is debited. Your account is credited for the entire amount in a single transaction. The issuer assumes the risk and establishes credit rates and fees that are charged to the customer. The customer pays the funding cost, which is a fee for paying in installments. In Brazil, a Crediario is a special type of issuer-funded installment payment plan that enables the customer to request information about the terms of the installment plan before approving the installment payments.
Merchant-Funded Installment Payments	The customer pays for goods or services using an installment plan agreed upon by you and the customer. The issuer controls how the customer's account is debited. Your account is credited periodically for partial amounts as the customer's account is debited. You assume the risk and establish the credit rate and fees that are charged to the customer.
Co-Branded Merchant Financed Installment Payments—Brazil Only	You and the issuer determine the terms for this kind of installment plan. The funding varies depending on the agreement between you, the issuer, and the customer. This funding method is available only for Mastercard installment payments in Brazil.
Issuer Merchant Co-Financed Installment Payments—Brazil Only	The issuer creates the installment plan. You and the issuer determine the service fees that the customer pays to you and the issuer. The acquirer is paid in full while the issuer is paid in installments by the customer. You or the customer pay the funding cost, which is a fee for paying in installments. This funding method is available only for Mastercard installment payments in Brazil.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Creating a TMS Token

When sending the initial CIT, you can create a TMS token to store the customer's credentials for the subsequent MITs. To create a TMS token, include the **processingInformation.actionTokenTypes** field in the authorization request. Set the field to one of these values based on the TMS token type you want to create:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "customer"
]
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token. Including a payment instrument in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "paymentInstrument"
]
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store a PAN. Including an instrument identifier

in subsequent MITs eliminates the need to include a PAN and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
    "instrumentIdentifier"
]
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Instrument Identifier, Payment Instrument, and Customer Identifier

You can also create multiple TMS token types in the same authorization. This example includes an instrument identifier, a payment instrument, and a customer token in the same authorization:

```
 "processingInformation": {
  "actionTokenTypes": [
    "instrumentIdentifier",
    "paymentInstrument",
    "customer"
]
```

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for CIT Installment Payments with TMS

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.actionList	Set the value to `TOKEN_CREATE`.
processingInformation.actionTokenTypes	Set to one or more of these values: • `customer` • `instrumentIdentifier` • `paymnentInstrument`
processingInformation.commerceIndicator	Set the value to `internet`, `MOTO`, or a payer authentication value.

Card-Specific Fields for Authorizing Initial Installment Payments

Use this required field if you are authorizing an initial installment payment using the card type referenced below.

Mastercard	**processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason** Set the value to `9`.

REST Example: CIT Installment Payment with TMS

Request

```
{
 "processingInformation": {
  "actionList": [
   "TOKEN_CREATE"
  ],
  "actionTokenTypes": [
   "instrumentIdentifier"
  ],
  "commerceIndicator": "internet"
 },
 "paymentInformation": {
  "card": {
   "number": "411111111111XXXX",
   "expirationMonth": "12",
   "expirationYear": "2031"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "4158880000"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6972267090226779103955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6972267090226779103955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6972267090226779103955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "id": "6972267090226779103955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
    "paymentAccountReferenceNumber": "V0010013022298169667504231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62506622XNMR6Q1Y",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-13T19:51:49Z",
  "tokenInformation": {
    "instrumentidentifierNew": false,
    "instrumentIdentifier": {
      "state": "ACTIVE",
      "id": "7010000000016241111"
    }
  }
}
```

Customer-Initiated Installment Payment with Enrollable Network Tokens

An installment payment is a single purchase of goods or services billed to a customer in multiple transactions over a period of time agreed to by you and the customer, and sometimes, the issuing bank. The agreement enables you to charge a specific amount at specified intervals. For customers, installment payments provide greater purchasing power and lower impact on their monthly budget. For you, offering installment payments at checkout can help increase the number of successfully completed purchases.

> **Important**
>
> Do not use this document if you are using the Installments service. When using the Installments service, Cybersource saves and stores payment credentials for installment transactions, ensuring compliance with COF best practices.

Using Enrollable Network Tokens

The Token Management Service can enroll certain network tokens, known as device tokens, into an instrument identifier token for future payments. Device tokens store and encrypt card-on-file information which enables customers to make quick and easy purchases using their mobile device. When authorizing a credentialed payment with a device token, you must create and store the device token in a TMS instrument identifier token. To do this, include the device token information in the **paymentInformation.tokenizedCard** fields and set the token creation fields to create an instrument identifier token.

Follow-on merchant-initiated transactions are performed using the created instrument identifier as the payment information. For more information about how to request a

merchant-initiated transaction, see *Merchant-Initiated Installment Payment with TMS* on page 405.
Device tokens are also known as digital payments, digital wallets, and tokenized cards.

Network Token Types

In your request, include the **processingInformation.paymentSolution** field to identify the device token type you are using, and set it to one of these possible values:

- `001`: Apple Pay
- `004`: Cybersource In-App Solution
- `005`: Masterpass
- `006`: Android Pay
- `007`: Chase Pay
- `008`: Samsung Pay
- `012`: Google Pay
- `014`: Mastercard credential-on-file (COF) payment network token
- `015`: Visa credential-on-file (COF) payment network token
- `027`: Click to Pay
- `visacheckout`: Visa Click to Pay.

Installment Payment Types

Visa Platform Connect enables you to process installment payments but does not have a role in setting the terms for the installment plan.
Visa Platform Connect enables you to process these types of installments payments:

Issuer-Funded Installment Payments	The customer pays for goods or services using an installment plan agreed upon by the customer and their issuing bank. The issuer controls how the customer's account is debited. Your account is credited for the entire amount in a single transaction. The issuer assumes the risk and establishes credit rates and fees that are charged to the customer. The customer pays the funding cost, which is a fee for paying in installments. In Brazil, a Crediario is a special type of issuer-funded installment payment plan that enables the customer to request information about the terms of the installment plan before approving the installment payments.
Merchant-Funded Installment Payments	The customer pays for goods or services using an installment plan agreed upon by you and the customer. The issuer controls how the customer's account is debited.

	Your account is credited periodically for partial amounts as the customer's account is debited. You assume the risk and establish the credit rate and fees that are charged to the customer.
Co-Branded Merchant Financed Installment Payments—Brazil Only	You and the issuer determine the terms for this kind of installment plan. The funding varies depending on the agreement between you, the issuer, and the customer. This funding method is available only for Mastercard installment payments in Brazil.
Issuer Merchant Co-Financed Installment Payments—Brazil Only	The issuer creates the installment plan. You and the issuer determine the service fees that the customer pays to you and the issuer. The acquirer is paid in full while the issuer is paid in installments by the customer. You or the customer pay the funding cost, which is a fee for paying in installments. This funding method is available only for Mastercard installment payments in Brazil.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for a CIT Installment Payment with Enrollable Network Tokens

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.tokenizedCard.expirationMonth

paymentInformation.tokenizedCard.expirationYear	
paymentInformation.tokenizedCard.number	
paymentInformation.tokenizedCard.transactionType	**Set the value to `1`.**
processingInformation.actionList	**Set the value to `TOKEN_CREATE`.**
processingInformation.actionTokenTypes	**Set the value to `instrumentIdentifier`.**
processingInformation.commerceIndicator	**Set the value to `internet`.**
processingInformation.paymentSolution	**Set to one of these possible values:** • `001`: Apple Pay • `004`: Cybersource In-App Solution • `005`: Masterpass • `006`: Android Pay • `007`: Chase Pay • `008`: Samsung Pay • `012`: Google Pay • `014`: Mastercard credential-on-file (COF) payment network token • `015`: Visa credential-on-file (COF) payment network token • `027`: Click to Pay • `visacheckout`: Visa Click to Pay.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

Example: CIT Installment Payments with Enrollable Network Tokens

Request

```
{
 "processingInformation": {
  "actionList": [
   "TOKEN_CREATE"
  ],
  "actionTokenTypes": [
   "instrumentIdentifier"
  ],
  "commerceIndicator": "internet",
  "paymentSolution": "001"
 },
  "paymentInformation": {
    "tokenizedCard": {
      "number": "4111111111111111",
      "expirationMonth": "02",
      "expirationYear": "2025",
      "transactionType": "1"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "USD"
    },
    "billTo": {
      "firstName": "John",
      "lastName": "Smith",
      "address1": "123 Happy St",
      "locality": "Austin",
      "administrativeArea": "TX",
      "postalCode": "78757",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": "444-4444-4444"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/7094060020036241803954/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/7094060020036241803954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/7094060020036241803954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1709406002076"
  },
  "id": "7094060020036241803954",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
```

```
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "60616704ST7Q27K2",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2024-03-02T19:00:02Z",
 "tokenInformation": {
  "instrumentidentifierNew": false,
  "instrumentIdentifier": {
   "state": "ACTIVE",
   "id": "7010000000016241111"
  }
 }
}
```

Merchant-Initiated Installment Payments with PAN

After the initial CIT installment payment, subsequent installment payments are merchant-initiated transactions (MITs).

Prerequisites

The first transaction in an installment payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Installment Payment Types

Visa Platform Connect enables you to process installment payments but does not have a role in setting the terms for the installment plan.
Visa Platform Connect enables you to process these types of installments payments:

Issuer-Funded Installment Payments	The customer pays for goods or services using an installment plan agreed upon by the customer and their issuing bank. The issuer controls how the customer's account is debited. Your account is credited for the entire amount in a single transaction. The issuer assumes the risk and establishes credit rates and fees that are charged to the customer. The customer pays the funding cost, which is a fee for paying in installments. In Brazil, a Crediario is a special type of issuer-funded installment payment plan that enables the customer to request information about the terms of the installment plan before approving the installment payments.
Merchant-Funded Installment Payments	The customer pays for goods or services using an installment plan agreed upon by you and the customer. The issuer controls how the customer's account is debited. Your account is credited periodically for partial amounts as the customer's account is debited. You assume the risk and establish the credit rate and fees that are charged to the customer.
Co-Branded Merchant Financed Installment Payments—Brazil Only	You and the issuer determine the terms for this kind of installment plan. The funding varies depending on the agreement between you, the issuer, and the customer. This funding method is available only for Mastercard installment payments in Brazil.
Issuer Merchant Co-Financed Installment Payments—Brazil Only	The issuer creates the installment plan. You and the issuer determine the service fees that the customer pays to you and the issuer. The acquirer is paid in full while the issuer is paid in installments by the customer. You or the customer pay the funding cost, which is a fee for paying in installments. This funding method is available only for Mastercard installment payments in Brazil.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing Merchant-Initiated Subsequent Installment Payments

Use these required fields to authorize merchant-initiated subsequent installment payments.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.. initiator. merchantInitiatedTransaction. previousTransactionID	American Express: set to the transaction ID from the original transaction. • Discover: set to the transaction ID from the original transaction. • Visa: set to the last successful transaction ID.
processingInformation. authorizationOptions. initiator. storedCredentialUsed	**Set the value to `true`.**
processingInformation. authorizationOptions. initiator. type	**Set the value to `merchant`.**

processingInformation. commerceIndicator **Set the value to `install`.**

Card-Specific Required Field for Retrieving Customer Credentials During a CIT

Discover

Discover requires the authorization amount from the original transaction in addition to the above required fields.

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction. originalAuthorizedAmount

Country-Specific Required Fields for Installment Payments with Mastercard or Visa Card
Include these country-specific required fields for installment payments using a Mastercard or Visa card, in addition to the required fields listed above.

Argentina

Include these required fields for payments using either a Mastercard or Visa card in Argentina.

installmentInformation.planType

installmentInformation.totalCount

processingInformation.commerceIndicator

Brazil

Include these required fields for payments using either a Mastercard or Visa card in Brazil.

buyerInformation.companyTaxId

buyerInformation.personalIdentification[].id

installmentInformation.planType

installmentInformation.totalCount

orderInformation.billTo.phoneNumber

processingInformation.loanOptions.type

Chile

Include these required fields for payments using either a Mastercard or Visa card in Chile.

installmentInformation.planType

installmentInformation.totalCount

processingInformation.commerceIndicator

Croatia

Include these required fields for payments using either a Mastercard or Visa card in Croatia.

installmentInformation.planType

merchantInformation.taxId

Georgia

Include these required fields for payments using either a Mastercard or Visa card in Georgia.

installmentInformation.amount

installmentInformation.firstInstallmentAmount

installmentInformation.monthlyInterestRate

installmentInformation.planType

installmentInformation.totalCount

Greece

Include these required fields for payments using either a Mastercard or Visa card in Greece.

installmentInformation.gracePeriodDuration

installmentInformation.gracePeriodDurationType

installmentInformation.planType

installmentInformation.totalCount

Mexico

Include these required fields for payments using either a Mastercard or Visa card in Mexico with Banco Nacional de México (Banamex) or BBVA México (Bancomer).

installmentInformation.amount

installmentInformation.paymentType

installmentInformation.planType

processingInformation.commerceIndicator

Paraguay

Include this required field for payments using either a Mastercard or Visa card in Paraguay.

installmentInformation.planType

Peru

Include this required field for payments using either a Mastercard or Visa card in Peru.

installmentInformation.planType

India-Specific Required Fields for Installment Payments

This section shows the required fields for Diners Club, Mastercard, and Visa in India.

Diners Club and Mastercard

Use these fields for authorizing an MIT installment payment when processing payments through Visa Platform Connect.

installmentInformation.amount	
installmentInformation.frequency	**Required only for the first MIT installment payment.**
installmentInformation.identifier	
installmentInformation.paymentType	
installmentInformation.sequence	
installmentInformation.validIndicator	

Visa

Use this field for authorizing a MIT installment payment when processing payments through Visa Platform Connect.

installmentInformation.identifier

REST Example: Authorizing Merchant-Initiated Subsequent Installment Payments

Request

```
{
  "processingInformation": {
    "commerceIndicator": "install",
    "authorizationOptions": {
      "initiator": {
        "storedCredentialUsed": "true",
        "type": "merchant",
        "merchantInitiatedTransaction": {
          "reason": "9",
          "previousTransactionId": "123456789619999",
          "originalAuthorizedAmount": "100"    //Discover Only
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "5554327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "79710341A39WTT5W",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

Merchant-Initiated Installment Payment with TMS

This section describes how to process a merchant-initiated installment payment using these TMS token types:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "customer": {
    "id": "07C9CA98022DA498E063A2598D0AA400"
  }
}
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token.

Including a payment instrument eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "paymentInstrument": {
    "id": "07CA24EF20F9E2C9E063A2598D0A8565"
  }
}
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store only a PAN. Including an instrument identifier eliminates the need to include a PAN and the previous transaction's ID.

```
"paymentInformation": {
  "instrumentIdentifier": {
    "id": "7010000000016241111"
  }
}
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Prerequisites

The first transaction in an installment payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Installment Payment Types

Visa Platform Connect enables you to process installment payments but does not have a role in setting the terms for the installment plan.

Visa Platform Connect enables you to process these types of installments payments:

Issuer-Funded Installment Payments

The customer pays for goods or services using an installment plan agreed upon by the customer and their issuing bank. The issuer controls how the customer's account is debited. Your account is credited for the entire amount in a single transaction. The issuer assumes the risk and establishes

	credit rates and fees that are charged to the customer. The customer pays the funding cost, which is a fee for paying in installments. In Brazil, a Crediario is a special type of issuer-funded installment payment plan that enables the customer to request information about the terms of the installment plan before approving the installment payments.
Merchant-Funded Installment Payments	The customer pays for goods or services using an installment plan agreed upon by you and the customer. The issuer controls how the customer's account is debited. Your account is credited periodically for partial amounts as the customer's account is debited. You assume the risk and establish the credit rate and fees that are charged to the customer.
Co-Branded Merchant Financed Installment Payments—Brazil Only	You and the issuer determine the terms for this kind of installment plan. The funding varies depending on the agreement between you, the issuer, and the customer. This funding method is available only for Mastercard installment payments in Brazil.
Issuer Merchant Co-Financed Installment Payments—Brazil Only	The issuer creates the installment plan. You and the issuer determine the service fees that the customer pays to you and the issuer. The acquirer is paid in full while the issuer is paid in installments by the customer. You or the customer pay the funding cost, which is a fee for paying in installments. This funding method is available only for Mastercard installment payments in Brazil.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for MIT Installment Payments with TMS

Include these Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.[tokentype].id	Where **[tokentype]** is the TMS token type you are using: • customer • instrumentIdentifier • paymentInstrument
processingInformation.commerceIndicator	Set the value to `install`.

Instrument Identifier Required Fields

If you are using the **paymentInformation.instrumentIdentifier.id** token, include these required fields in addition to the required fields listed above.

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

Card-Specific Required Field for Retrieving Customer Credentials During a CIT

Discover

Discover requires the authorization amount from the original transaction in addition to the above required fields.

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.originalAuthorizedAmount

Country-Specific Required Fields for Installment Payments with Mastercard or Visa Card
Include these country-specific required fields for installment payments using a Mastercard or Visa card, in addition to the required fields listed above.

Argentina

Include these required fields for payments using either a Mastercard or Visa card in Argentina.

installmentInformation.planType

installmentInformation.totalCount

processingInformation.commerceIndicator

Brazil

Include these required fields for payments using either a Mastercard or Visa card in Brazil.

buyerInformation.companyTaxId

buyerInformation.personalIdentification[].id

installmentInformation.planType

installmentInformation.totalCount

orderInformation.billTo.phoneNumber

processingInformation.loanOptions.type

Chile

Include these required fields for payments using either a Mastercard or Visa card in Chile.

installmentInformation.planType

installmentInformation.totalCount

processingInformation.commerceIndicator

Croatia

Include these required fields for payments using either a Mastercard or Visa card in Croatia.

installmentInformation.planType

merchantInformation.taxId

Georgia

Include these required fields for payments using either a Mastercard or Visa card in Georgia.

installmentInformation.amount

installmentInformation.firstInstallmentAmount

installmentInformation.monthlyInterestRate

installmentInformation.planType

installmentInformation.totalCount

Greece

Include these required fields for payments using either a Mastercard or Visa card in Greece.

installmentInformation.gracePeriodDuration

installmentInformation.gracePeriodDurationType

installmentInformation.planType

installmentInformation.totalCount

Mexico

Include these required fields for payments using either a Mastercard or Visa card in Mexico with Banco Nacional de México (Banamex) or BBVA México (Bancomer).

installmentInformation.amount

installmentInformation.paymentType

installmentInformation.planType

processingInformation.commerceIndicator

Paraguay

Include this required field for payments using either a Mastercard or Visa card in Paraguay.

installmentInformation.planType

Peru

Include this required field for payments using either a Mastercard or Visa card in Peru.

installmentInformation.planType

India-Specific Required Fields for Installment Payments

This section shows the required fields for Diners Club, Mastercard, and Visa in India.

Diners Club and Mastercard

Use these fields for authorizing an MIT installment payment when processing payments through Visa Platform Connect.

installmentInformation.amount	
installmentInformation.frequency	**Required only for the first MIT installment payment.**
installmentInformation.identifier	
installmentInformation.paymentType	

installmentInformation.sequence

installmentInformation.validIndicator

Visa

Use this field for authorizing a MIT installment payment when processing payments through Visa Platform Connect.

installmentInformation.identifier

Example: MIT with TMS Instrument Identifier Token

Request

```
{
 "processingInformation": {
  "commerceIndicator": "install"
 },
 "paymentInformation": {
  "card": {
   "expirationMonth": "12",
   "expirationYear": "2031"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "4158880000"
  }
 }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "79710341A39WTT5W",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

Recurring Payments

A recurring payment is a credentials-on-file (COF) transaction in a series of payments that you bill to a customer for a fixed amount at regular intervals that do not exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals. Recurring payments are also known as subscriptions.

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 439 and *Mastercard Subscription Payments* on page 447.

Recurring Billing Service for Recurring Payments

Important

Do not use this document for the Recurring Billing service.

Use the *Recurring Billing Developer Guide*. When you use the Recurring Billing service, Cybersource saves and stores payment credentials for recurring transactions, ensuring compliance with COF best practices.

Customer-Initiated Recurring Payment with PAN

A recurring payment is a credentials-on-file (COF) transaction in a series of payments that you bill to a customer at a fixed amount, at regular intervals that do not exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Visa

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 439 and *Mastercard Subscription Payments* on page 447.

Recurring Billing Service for Recurring Payments

Important

Do not use this document for the Recurring Billing service.

> Use the *Recurring Billing Developer Guide*. When you use the Recurring Billing service, Cybersource saves and stores payment credentials for recurring transactions, ensuring compliance with COF best practices.

Address Verification Service for Recurring Payments

If your processor supports the Address Verification Service (AVS), then the AVS should verify every authorization request. Cybersource recommends checking the AVS's results for the first recurring payment to ensure that the payment information is accurate and to reduce the risk of fraud.

You must determine how to handle the AVS results for any subsequent recurring payments that are not the same as the already-verified billing address information from the first recurring payment.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Successful Response

You must store the network transaction ID from the successful response message to include in subsequent MIT authorization requests in order to associate the CIT to the MIT. The network transaction ID is the **processorInformation.networkTransactionId** field value.

Store the network transaction ID, which is the **processorInformation.networkTransactionId** field value, from the successful response message. You must include the network transaction ID in subsequent MIT authorization requests in order to associate the CIT to the MIT.

Required Fields for Authorizing a Customer-Initiated Recurring Payment with PAN

Use these required fields to request an initial customer-initiated recurring payment.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation. authorizationOptions. initiator. credentialStoredOnFile	Set the value to `true`.
processingInformation. authorizationOptions. initiator. type	Set the value to `customer`.
processingInformation. commerceIndicator	Set the value to `internet`, a payer authentication value, or `MOTO`.
processingInformation. recurringOptions. firstRecurringPayment	Set the value to `true`.

REST Example: Authorizing a Customer-Initiated Recurring Payment with a PAN

Request

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "credentialStoredOnFile": "true",
        "type": "customer"
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "5554327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6528187198946076303004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1652818719876"
  },
  "id": "6528187198946076303004",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "63165088Z3AHV91G",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-17T20:18:40Z"
}
```

Customer-Initiated Recurring Payment with TMS

A recurring payment is a credentials-on-file (COF) transaction in a series of payments that you bill to a customer at a fixed amount, at regular intervals that do not exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Visa

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 439 and *Mastercard Subscription Payments* on page 447.

Recurring Billing Service for Recurring Payments

> **Important**
>
> Do not use this document for the Recurring Billing service.
> Use the *Recurring Billing Developer Guide*. When you use the Recurring Billing service, Cybersource saves and stores payment credentials for recurring transactions, ensuring compliance with COF best practices.

Creating a TMS Token

When sending the initial CIT, you can create a TMS token to store the customer's credentials for the subsequent MITs. To create a TMS token, include the **processingInformation.actionTokenTypes** field in the authorization request. Set the field to one of these values based on the TMS token type you want to create:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
    "customer"
]
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token. Including a payment instrument in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
    "paymentInstrument"
]
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store a PAN. Including an instrument identifier in subsequent MITs eliminates the need to include a PAN and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
    "instrumentIdentifier"
]
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Instrument Identifier, Payment Instrument, and Customer Identifier

You can also create multiple TMS token types in the same authorization. This example includes an instrument identifier, a payment instrument, and a customer token in the same authorization:

```
 "processingInformation": {
   "actionTokenTypes": [
     "instrumentIdentifier",
    "paymentInstrument",
    "customer"
]
```

Address Verification Service for Recurring Payments

If your processor supports the Address Verification Service (AVS), then the AVS should verify every authorization request. Cybersource recommends checking the AVS's results for the first recurring payment to ensure that the payment information is accurate and to reduce the risk of fraud.

You must determine how to handle the AVS results for any subsequent recurring payments that are not the same as the already-verified billing address information from the first recurring payment.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`

Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Customer-Initiated Recurring Payment with TMS

Use these required fields to request a customer-initiated recurring payment with TMS.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.actionList	Set the value to `TOKEN_CREATE`.
processingInformation.actionTokenTypes	Set to one or more of these values: • `customer` • `instrumentIdentifier`

	- paymentInstrument
processingInformation.commerceIndicator	Set the value to internet, MOTO, or a payer authentication value.
processingInformation.recurringOptions.firstRecurringPayment	Set the value to true.

REST Example: Authorizing a Customer-Initiated Recurring Payment with TMS

Request

```
{
  "processingInformation": {
    "actionList": [
      "TOKEN_CREATE"
    ],
    "actionTokenTypes": [
      "customer"
    ],
    "commerceIndicator": "internet",
    "recurringOptions": {
      "firstRecurringPayment": true
    }
  },
  "paymentInformation": {
    "card": {
      "number": "4111111111111111",
      "expirationMonth": "12",
      "expirationYear": "2031"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "USD"
    },
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "1 Market St",
      "locality": "san francisco",
      "administrativeArea": "CA",
      "postalCode": "94105",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": ""
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
   "href": "/pts/v2/payments/6976858134106105703954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976858134106105703954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976858134106105703954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697685813462"
 },
 "id": "6976858134106105703954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62698397FNN143CC",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T03:23:33Z",
 "tokenInformation": {
  "customer": {
   "id": "080A3A742BF87171E063A2598D0AEABE"
  }
 }
}
```

Customer-Initiated Recurring Payment with Enrollable Network Tokens

A recurring payment is a credentials-on-file (COF) transaction in a series of payments that you bill to a customer at a fixed amount, at regular intervals that do not exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 439 and *Mastercard Subscription Payments* on page 447.

Recurring Billing Service for Recurring Payments

> **Important**
>
> Do not use this document for the Recurring Billing service.
>
> Use the *Recurring Billing Developer Guide*. When you use the Recurring Billing service, Cybersource saves and stores payment credentials for recurring transactions, ensuring compliance with COF best practices.

Using Enrollable Network Tokens

The Token Management Service can enroll certain network tokens, known as device tokens, into an instrument identifier token for future payments. Device tokens store and encrypt card-on-file information which enables customers to make quick and easy purchases using their mobile device. When authorizing a credentialed payment with a device token, you must create and store the device token in a TMS instrument identifier token. To do this, include the device token information in the **paymentInformation.tokenizedCard** fields and set the token creation fields to create an instrument identifier token.

Follow-on merchant-initiated transactions are performed using the created instrument identifier as the payment information. For more information about how to request a merchant-initiated transaction, see *Merchant-Initiated Recurring Payments with TMS* on page 431.

Device tokens are also known as digital payments, digital wallets, and tokenized cards.

Network Token Types

In your request, include the **processingInformation.paymentSolution** field to identify the device token type you are using, and set it to one of these possible values:

- `001`: Apple Pay
- `004`: Cybersource In-App Solution
- `005`: Masterpass
- `006`: Android Pay
- `007`: Chase Pay

- `008`: Samsung Pay
- `012`: Google Pay
- `014`: Mastercard credential-on-file (COF) payment network token
- `015`: Visa credential-on-file (COF) payment network token
- `027`: Click to Pay
- `visacheckout`: Visa Click to Pay.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Customer-Initiated Recurring Payments with Enrollable Network Tokens

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.tokenizedCard.expirationMonth	
paymentInformation.tokenizedCard.expirationYear	
paymentInformation.tokenizedCard.number	
paymentInformation.tokenizedCard.transactionType	**Set the value to `1`.**
processingInformation.actionList	**Set the value to `TOKEN_CREATE`.**
processingInformation.actionTokenTypes	**Set the value to `instrumentIdentifier`.**
processingInformation.commerceIndicator	**Set the value to `internet`, `MOTO`, or a payer authentication value.**
processingInformation.paymentSolution	**Set to one of these possible values:** • `001`: Apple Pay • `004`: Cybersource In-App Solution • `005`: Masterpass

- `006`: Android Pay
- `007`: Chase Pay
- `008`: Samsung Pay
- `012`: Google Pay
- `014`: Mastercard credential-on-file (COF) payment network token
- `015`: Visa credential-on-file (COF) payment network token
- `027`: Click to Pay
- `visacheckout`: Visa Click to Pay.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 300.

REST Example: Authorizing a Customer-Initiated Recurring Payment with Enrollable Network Tokens

Request

```
{
 "processingInformation": {
  "actionList": [
   "TOKEN_CREATE"
  ],
  "actionTokenTypes": [
   "instrumentIdentifier"
  ],
  "commerceIndicator": "internet",
  "paymentSolution": "001"
 },
 "paymentInformation": {
  "tokenizedCard": {
   "number": "4111111111111111",
   "expirationMonth": "02",
   "expirationYear": "2025",
   "transactionType": "1"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Smith",
   "address1": "123 Happy St",
   "locality": "Austin",
   "administrativeArea": "TX",
   "postalCode": "78757",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "444-4444-4444"
  }
 }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/7094060020036241803954/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/7094060020036241803954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/7094060020036241803954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1709406002076"
  },
  "id": "7094060020036241803954",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "60616704ST7Q27K2",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2024-03-02T19:00:02Z",
 "tokenInformation": {
  "instrumentidentifierNew": false,
  "instrumentIdentifier": {
   "state": "ACTIVE",
   "id": "7010000000016241111"
  }
 }
}
```

Merchant-Initiated Recurring Payments with PAN

After the initial recurring payment (CIT), subsequent recurring payments are merchant-initiated transactions (MITs).

Prerequisites

The first transaction in a recurring payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the customer's credentials, you must get their consent to store their private information. This is also known as establishing a relationship with the customer.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Visa

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 439 and *Mastercard Subscription Payments* on page 447.

Address Verification Service for Recurring Payments

If your processor supports the Address Verification Service (AVS), then the AVS should verify every authorization request. Cybersource recommends checking the AVS's results for the first recurring payment to ensure that the payment information is accurate and to reduce the risk of fraud.
You must determine how to handle the AVS results for any subsequent recurring payments that are not the same as the already-verified billing address information from the first recurring payment.

Replacing Expiration Dates

If the customer's card-on-file is going to expire before a scheduled subsequent recurring payment, your processor may allow you to replace the expiration date with the date 12/2099.

> **Important**
>
> Do not replace a card's expiration date if the card is not expired.

Using this replacement expiration date does not guarantee a successful authorization request. It is your responsibility to know if your processor supports this feature. Not all issuing banks support the 12/2099 expiration date and may decline the authorization request.
To include this date in the authorization request, use these fields and values.

paymentInformation.card.expirationMonth	**Set to** 12**.**
paymentInformation.card.expirationYear	**Set to** 99**.**

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Merchant-Initiated Recurring Payment

Use these required fields to authorize subsequent recurring payments.

authorizationOptions. initiator. merchantInitiatedTransaction. agreementId	**Required for the first MIT recurring payment and subsequent MIT recurring payments if your business is located in Saudi Arabia.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	

paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation. card. number	
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. previousTransactionID	**For Discover and American Express cards, use the transaction ID from the original transaction. For Visa, use the last successful transaction ID.**
processingInformation. authorizationOptions. initiator. storedCredentialUsed	**Set the value to** `true`**.**
processingInformation. authorizationOptions. initiator. type	**Set the value to** `merchant`**.**
processingInformation. commerceIndicator	**Set the value to** `recurring`**.**

Card-Specific Required Fields for Authorizing Subsequent Recurring Payments

Some card companies require additional information when making authorizations with stored credentials.

Discover

Include the authorization amount from the original transaction in this field:

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction. originalAuthorizedAmount

Mastercard

Mastercard supports subscription and standing order payments instead of recurring payments.
See *Mastercard Subscription Payments* on page 447 and *Mastercard Standing Order Payments* on page 439.

Country-Specific Required Fields for Authorizing Subsequent Recurring Payments

Include these country-specific required fields for a successful merchant-initiated authorization.

India

These fields are required only with Diners Club in India or with an India-issued card, and you are processing payments through Visa Platform Connect.

installmentInformation.amount

installmentInformation.frequency

installmentInformation.identifier

installmentInformation.paymentType

installmentInformation.sequence

installmentInformation.validationIndicator

Saudi Arabia

These fields are required only if your business is located in Saudi Arabia and you are processing payments through Visa Platform Connect.

authorizationOptions.initiator.merchantInitiatedTransaction.agreementId

recurringPaymentInformation.amountType

REST Example: Authorizing a Merchant-Initiated Recurring Payment

Request

```
{
  "processingInformation": {
    "commerceIndicator": "recurring",
    "authorizationOptions": {
      "initiator": {
        "storedCredentialUsed": "true",
        "type": "merchant",
        "merchantInitiatedTransaction": {
          "previousTransactionId": "123456789619999",
          "originalAuthorizedAmount": "100"    //Discover Only
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "5554327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
   "reconciliationId": "79710341A39WTT5W",
   "status": "AUTHORIZED",
   "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

Merchant-Initiated Recurring Payments with TMS

After the customer-initiated recurring payment, you can send merchant-initiated recurring payments using one or more TMS token types:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "customer": {
    "id": "07C9CA98022DA498E063A2598D0AA400"
  }
}
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token.

Including a payment instrument eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "paymentInstrument": {
    "id": "07CA24EF20F9E2C9E063A2598D0A8565"
  }
}
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store only a PAN. Including an instrument identifier eliminates the need to include a PAN and the previous transaction's ID.

```
"paymentInformation": {
 "instrumentIdentifier": {
  "id": "7010000000016241111"
 }
}
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Prerequisites

The first transaction in a recurring payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the customer's credentials, you must get their consent to store their private information. This is also known as establishing a relationship with the customer.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 439 and *Mastercard Subscription Payments* on page 447.

Address Verification Service for Recurring Payments

If your processor supports the Address Verification Service (AVS), then the AVS should verify every authorization request. Cybersource recommends checking the AVS's results for the first recurring payment to ensure that the payment information is accurate and to reduce the risk of fraud.

You must determine how to handle the AVS results for any subsequent recurring payments that are not the same as the already-verified billing address information from the first recurring payment.

Replacing Expiration Dates

If the customer's card-on-file is going to expire before a scheduled subsequent recurring payment, your processor may allow you to replace the expiration date with the date 12/2099.

> **Important**
>
> Do not replace a card's expiration date if the card is not expired.

Using this replacement expiration date does not guarantee a successful authorization request. It is your responsibility to know if your processor supports this feature. Not all issuing banks support the 12/2099 expiration date and may decline the authorization request.

To include this date in the authorization request, use these fields and values.

paymentInformation.card.expirationMonth	**Set to** 12**.**
paymentInformation.card.expirationYear	**Set to** 99**.**

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Merchant-Initiated Recurring Payments with TMS

Use these required fields to authorize subsequent recurring payments.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.[tokentype].id	**Where [tokentype] is the TMS token type you are using:** • **customer** • **instrumentIdentifier** • **paymentInstrument**
processingInformation.commerceIndicator	**Set the value to** recurring**.**

Instrument Identifier Required Fields

If you are using the **paymentInformation.instrumentIdentifier.id** token, include these required fields in addition to the required fields listed above.

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

Card-Specific Field

Some card companies require additional fields when making authorizations with stored credentials. Include this field if you are using these card types:

Discover	**processingInformation.authorizationOptions.initiator. merchantInitiatedTransaction.originalAuthorizedAmou**
Mastercard	Mastercard supports subscription and standing order payments instead of recurring payments. See *Mastercard Subscription Payments* on page 447 and *Mastercard Standing Order Payments* on page 439.

Country-Specific Field

Some countries require additional fields in order to process an authorization. Include this field if your business is located in this country:

Saudi Arabia	**authorizationOptions.initiator.merchantInitiatedTransa** **Required for the first MIT recurring payment and subsequent MIT recurring payments.**

REST Example: Authorizing a Merchant-Initiated Recurring Payment with a TMS Instrument Identifier

Request

```
{
  "processingInformation": {
    "commerceIndicator": "recurring"
  },
  "paymentInformation": {
    "card": {
      "expirationMonth": "12",
      "expirationYear": "2025"
    },
    "instrumentIdentifier": {
      "id": "4111xxxxxxxxxxxx"
    }
  },
  "orderInformation": {
    "amountDetails": {
```

```
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Smith",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "4158880000"
  }
 }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "79710341A39WTT5W",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

REST Example: Authorizing a Merchant-Initiated Recurring Payment with TMS Payment Instrument

Request

```
{
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "processingInformation": {
  "commerceIndicator": "recurring"
 },
 "paymentInformation": {
  "paymentInstrument": {
   "id": "07DB0915C20F2DDBE063A2598D0A6F26"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6974839908106304103955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6974839908106304103955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6974839908106304103955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "id": "6974839908106304103955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "07DB0915C20F2DDBE063A2598D0A6F26"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62599243NNMR6324",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-16T19:19:51Z"
```

```
}
```

REST Example: Authorizing a Merchant-Initiated Recurring Payment with a TMS Customer Token

Request

```
{
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "processingInformation": {
    "commerceIndicator": "recurring"
  },
  "paymentInformation": {
    "customer": {
      "id": "07DB50E35AE11DA2E063A2598D0A9995"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "USD"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6974846967476340503955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6974846967476340503955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6974846967476340503955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "id": "6974846967476340503955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
```

```
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62599950BNN133LK",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-16T19:31:36Z"
}
```

Mastercard Standing Order Payments

A standing order payment is a recurring COF transaction that is a variable amount at a regular interval, such as a utility bill, not to exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard Initial CIT Standing Order Payment

The first transaction in a standing order payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`

Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Successful Response

You must store the network transaction ID from the successful response message to include in subsequent MIT authorization requests in order to associate the CIT to the MIT. The network transaction ID is the **processorInformation.networkTransactionId** field value. Store the network transaction ID, which is the **processorInformation.networkTransactionId** field value, from the successful response message. You must include the network transaction ID in subsequent MIT authorization requests in order to associate the CIT to the MIT.

Required Fields for Authorizing Initial CIT Standing Order Payments

Use these required fields to authorize initial customer-initated standing order payments.

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation. authorizationOptions. initiator. credentialStoredOnFile	**Set the value to `true`.**
processingInformation. authorizationOptions. initiator.type	**Set the value to `customer`.**
processingInformation. commerceIndicator	**Set the value to `internet`, `MOTO`, or a payer authentication value.**
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. reason	**Set the value to `8`.**

REST Example: Authorizing Initial CIT Standing Order Payments

Request

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "credentialStoredOnFile": "true",
        "type": "customer",
        "merchantInitiatedTransaction": {
          "reason": "8"
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "5554327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "5555xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "79710341A39WTT5W",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

Mastercard Initial CIT Standing Order Payment with TMS

The first transaction in a standing order payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Creating a TMS Token

When sending the initial CIT, you can create a TMS token to store the customer's credentials for the subsequent MITs. To create a TMS token, include the **processingInformation.actionTokenTypes** field in the authorization request. Set the field to one of these values based on the TMS token type you want to create:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "customer"
 ]
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token. Including a payment instrument in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "paymentInstrument"
 ]
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store a PAN. Including an instrument identifier in subsequent MITs eliminates the need to include a PAN and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
   "instrumentIdentifier"
]
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Instrument Identifier, Payment Instrument, and Customer Identifier

You can also create multiple TMS token types in the same authorization. This example includes an instrument identifier, a payment instrument, and a customer token in the same authorization:

```
"processingInformation": {
  "actionTokenTypes": [
     "instrumentIdentifier",
     "paymentInstrument",
     "customer"
]
```

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing Initial CIT Standing Order Payments with TMS

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

processingInformation.actionList	Set the value to `TOKEN_CREATE`
processingInformation.actionTokenTypes	Set to one or more of these values: • `customer` • `instrumentIdentifier` • `paymentInstrument`
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	Set the value to `8`.
processingInformation.commerceIndicator	Set the value to `internet`, `MOTO`, or a payer authentication value.

REST Example: Authorizing Initial CIT Standing Order Payments with TMS

Request

```
{
  "processingInformation": {
    "actionList": ["TOKEN_CREATE"],
    "actionTokenTypes": ["customer"],
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "merchantInitiatedTransaction": {
          "reason": "8"
        }
      }
    }
  },
  "paymentInformation": {
    "card": {
      "number": "555555555555xxxx",
      "expirationMonth": "12",
      "expirationYear": "2031"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    },
    "billTo": {
      "firstName": "John",
      "lastName": "Smith",
      "address1": "123 Happy St",
      "locality": "Sunnyville",
      "administrativeArea": "CA",
      "postalCode": "55555",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": "444-4444-4444"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/7064959411486706503954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7064959411486706503954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/7064959411486706503954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1706495941197"
 },
 "id": "7064959411486706503954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "100.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "002"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "002"
  },
  "card": {
   "type": "002"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "authIndicator": "1",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "680915409RRMGL34",
 "status": "AUTHORIZED",
  "submitTimeUtc": "2024-01-29T02:39:01Z",
  "tokenInformation": {
    "customer": {
      "id": "100D6CDA178DD64DE063A2598D0AD3D5"
    }
  }
}
```

Mastercard Subscription Payments

A subscription payment is a recurring COF transaction that is processed at a fixed amount at regular intervals not to exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard CIT Initial Subscription Payment

The first transaction in a subscription payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Successful Response

You must store the network transaction ID from the successful response message to include in subsequent MIT authorization requests in order to associate the CIT to the MIT. The network transaction ID is the **processorInformation.networkTransactionId** field value. Store the network transaction ID, which is the **processorInformation.networkTransactionId** field value, from the successful response message. You must include the network transaction ID in subsequent MIT authorization requests in order to associate the CIT to the MIT.

Required Fields for Authorizing CIT Initial Subscription Payments

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

processingInformation.authorizationOptions.initiator.credentialStoredOnFile Set the value to `true`.

processingInformation.authorizationOptions.initiator.type Set the value to `customer`.

processingInformation.commerceIndicator Set the value to `recurring`.

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason Set the value to `7`.

REST Example: Authorizing Initial CIT Subscription Payments

Request

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "type": "customer",
        "credentialStoredOnFile": "true",
        "merchantInitiatedTransaction": {
          "reason": "7"
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "5554327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "79710341A39WTT5W",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

Mastercard CIT Initial Subscription Payment with TMS

The first transaction in a subscription payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Creating a TMS Token

When sending the initial CIT, you can create a TMS token to store the customer's credentials for the subsequent MITs. To create a TMS token, include the **processingInformation.actionTokenTypes** field in the authorization request. Set the field to one of these values based on the TMS token type you want to create:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
   "customer"
]
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token. Including a payment instrument in subsequent MITs eliminates the need to include billing information, card

information, and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "paymentInstrument"
]
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store a PAN. Including an instrument identifier in subsequent MITs eliminates the need to include a PAN and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "instrumentIdentifier"
]
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Instrument Identifier, Payment Instrument, and Customer Identifier

You can also create multiple TMS token types in the same authorization. This example includes an instrument identifier, a payment instrument, and a customer token in the same authorization:

```
 "processingInformation": {
   "actionTokenTypes": [
      "instrumentIdentifier",
      "paymentInstrument",
      "customer"
]
```

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for Authorizing CIT Initial Subscription Payments with TMS

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.actionList	Set the value to `TOKEN_CREATE`
processingInformation.actionTokenTypes	Set to one or more of these values: • `customer` • `instrumentIdentifier` • `paymentInstrument`
processingInformation.commerceIndicator	Set the value to `recurring`.
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	Set the value to `7`.

REST Example: Authorizing Initial CIT Subscription Payments with TMS

Request

```
{
 "processingInformation": {
  "actionList": ["TOKEN_CREATE"],
  "actionTokenTypes": ["customer"],
  "commerceIndicator": "recurring",
  "authorizationOptions": {
   "initiator": {
    "merchantInitiatedTransaction": {
     "reason": "7"
    }
   }
  }
 },
 "paymentInformation": {
  "card": {
   "number": "555555555555xxxx",
   "expirationMonth": "12",
   "expirationYear": "2031"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "100.00",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Smith",
   "address1": "123 Happy St",
   "locality": "Sunnyville",
   "administrativeArea": "CA",
   "postalCode": "55555",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "444-4444-4444"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/7064946846256410103954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7064946846256410103954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/7064946846256410103954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1706494684667"
 },
 "id": "7064946846256410103954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "100.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "002"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "002"
  },
  "card": {
   "type": "002"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "authIndicator": "1",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "68091233JRRDUQ34",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2024-01-29T02:18:04Z",
 "tokenInformation": {
  "customer": {
   "id": "100D1DC40CC7C803E063A2598D0A29BD"
  }
 }
}
```

Unscheduled COF Payments

An unscheduled credentials-on-file (COF) transaction uses stored payment information for a fixed or variable amount that does not occur regularly. An account top-up is one kind of unscheduled COF.

Customer-Initiated Unscheduled COF Payment with PAN

An unscheduled credentials-on-file (COF) transaction uses stored payment information for a fixed or variable amount that does not occur regularly. An account top-up is one kind of unscheduled COF.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Successful Response

You must store the network transaction ID from the successful response message to include in subsequent MIT authorization requests in order to associate the CIT to the MIT. The network transaction ID is the **processorInformation.networkTransactionId** field value. Store the network transaction ID, which is the **processorInformation.networkTransactionId** field value, from the successful response message. You must include the network transaction ID in subsequent MIT authorization requests in order to associate the CIT to the MIT.

Required Fields for a Customer-Initiated Unscheduled COF Payment with PAN

These fields are required in a subsequent authorization request for an initial unscheduled COF payment:

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation. authorizationOptions. initiator. credentialStoredOnFile	**Set the value to `true`.**
processingInformation. authorizationOptions. initiator. type	**Set the value to `customer`.**
processingInformation. commerceIndicator	**Set the value to `internet`, `MOTO`, or a payer authentication value.**

REST Example: Customer-Initiated Unscheduled COF Payment with PAN

Request

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "credentialStoredOnFile": "true",
        "type": "customer"
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "5554327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6528187198946076303004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
    },
    "clientReferenceInformation": {
        "code": "1652818719876"
    },
    "id": "6528187198946076303004",
    "orderInformation": {
        "amountDetails": {
            "authorizedAmount": "100.00",
            "currency": "USD"
        }
    },
    "paymentAccountInformation": {
        "card": {
            "type": "001"
        }
    },
    "paymentInformation": {
        "tokenizedCard": {
            "type": "001"
        },
        "card": {
            "type": "001"
        }
    },
    "pointOfSaleInformation": {
        "terminalId": "111111"
    },
    "processorInformation": {
        "approvalCode": "888888",
        "networkTransactionId": "123456789619999",
        "transactionId": "123456789619999",
        "responseCode": "100",
        "avs": {
            "code": "X",
            "codeRaw": "I1"
        }
    },
    "reconciliationId": "63165088Z3AHV91G",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2022-05-17T20:18:40Z"
}
```

Customer-Initiated Unscheduled COF Payments with TMS

An unscheduled credentials-on-file (COF) transaction uses stored payment information for a fixed or variable amount that does not occur regularly. An account top-up is one kind of unscheduled COF.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard

- Visa

Creating a TMS Token

When sending the initial CIT, you can create a TMS token to store the customer's credentials for the subsequent MITs. To create a TMS token, include the **processingInformation.actionTokenTypes** field in the authorization request. Set the field to one of these values based on the TMS token type you want to create:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "customer"
 ]
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token. Including a payment instrument in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
   "actionTokenTypes": [
     "paymentInstrument"
 ]
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store a PAN. Including an instrument identifier in subsequent MITs eliminates the

	need to include a PAN and the previous transaction's ID.
	`"processingInformation": {`
	`  "actionTokenTypes": [`
	`    "instrumentIdentifier"`
	`]`
	For more information about this TMS token type, see ***Instrument Identifier Token*** **in the Token Management Service Developer Guide.**

Instrument Identifier, Payment Instrument, and Customer Identifier	**You can also create multiple TMS token types in the same authorization. This example includes an instrument identifier, a payment instrument, and a customer token in the same authorization:**
	`"processingInformation": {`
	`  "actionTokenTypes": [`
	`    "instrumentIdentifier",`
	`    "paymentInstrument",`
	`    "customer"`
	`]`

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for CIT Unscheduled COF Payments with TMS

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.actionList	Set the value to `TOKEN_CREATE`
processingInformation.actionTokenTypes	Set to one or more of these values: • `customer` • `instrumentIdentifier` • `paymnentInstrument`
processingInformation.commerceIndicator	Set the value to `internet`, `MOTO`, or a payer authentication value.

REST Example: Initial CIT Unscheduled COF Payment in TMS

Request

```
{
 "processingInformation": {
  "actionList": [
   "TOKEN_CREATE"
  ],
  "actionTokenTypes": [
   "customer"
  ],
  "commerceIndicator": "internet"
 },
 "paymentInformation": {
  "card": {
   "number": "4111111111111111",
   "expirationMonth": "12",
   "expirationYear": "2031"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "444-4444-4444"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976866073586557303955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976866073586557303955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976866073586557303955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697686607441"
 },
 "id": "6976866073586557303955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62699023FNN143DG",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T03:36:47Z",
 "tokenInformation": {
  "customer": {
   "id": "080A6C3842C72DCBE063A2598D0AA98B"
  }
 }
}
```

Customer-Initiated Unscheduled COF Payment with Enrollable Network Tokens

An unscheduled credentials-on-file (COF) transaction uses stored payment information for a fixed or variable amount that does not occur regularly. An account top-up is one kind of unscheduled COF.

Using Enrollable Network Tokens

The Token Management Service can enroll certain network tokens, known as device tokens, into an instrument identifier token for future payments. Device tokens store and encrypt card-on-file information which enables customers to make quick and easy purchases using their mobile device. When authorizing a credentialed payment with a device token, you must create and store the device token in a TMS instrument identifier token. To do this, include the device token information in the **paymentInformation.tokenizedCard** fields and set the token creation fields to create an instrument identifier token.

Follow-on merchant-initiated transactions are performed using the created instrument identifier as the payment information. For more information about how to request a merchant-initiated transaction, see *Merchant-Initiated Unscheduled COF Payments with TMS* on page 470.

Device tokens are also known as digital payments, digital wallets, and tokenized cards.

Network Token Types

In your request, include the **processingInformation.paymentSolution** field to identify the device token type you are using, and set it to one of these possible values:

- `001`: Apple Pay
- `004`: Cybersource In-App Solution
- `005`: Masterpass
- `006`: Android Pay
- `007`: Chase Pay
- `008`: Samsung Pay
- `012`: Google Pay
- `014`: Mastercard credential-on-file (COF) payment network token
- `015`: Visa credential-on-file (COF) payment network token
- `027`: Click to Pay
- `visacheckout`: Visa Click to Pay.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Required Fields for CIT Unscheduled COF Payment with Enrollable Network Tokens

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.tokenizedCard.expirationMonth	
paymentInformation.tokenizedCard.expirationYear	
paymentInformation.tokenizedCard.number	
paymentInformation.tokenizedCard.transactionType	**Set the value to `1`.**
processingInformation.actionList	**Set the value to `TOKEN_CREATE`.**
processingInformation.actionTokenTypes	**Set the value to `instrumentIdentifier`.**
processingInformation.commerceIndicator	**Set the value to `internet`, `MOTO`, or a payer authentication value.**
processingInformation.paymentSolution	**Set to one of these possible values:** • `001`: Apple Pay • `004`: Cybersource In-App Solution • `005`: Masterpass • `006`: Android Pay • `007`: Chase Pay • `008`: Samsung Pay • `012`: Google Pay • `014`: Mastercard credential-on-file (COF) payment network token

- `015`: Visa credential-on-file (COF) payment network token
- `027`: Click to Pay
- `visacheckout`: Visa Click to Pay.

REST API Example: CIT Unscheduled COF Payment with Enrollable Network Tokens

Request

```
{
  "processingInformation": {
    "actionList": [
      "TOKEN_CREATE"
    ],
    "actionTokenTypes": [
      "instrumentIdentifier"
    ],
    "commerceIndicator": "internet",
    "paymentSolution": "001"
  },
  "paymentInformation": {
    "tokenizedCard": {
      "number": "4111111111111111",
      "expirationMonth": "02",
      "expirationYear": "2025",
      "transactionType": "1"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "USD"
    },
    "billTo": {
      "firstName": "John",
      "lastName": "Smith",
      "address1": "123 Happy St",
      "locality": "Austin",
      "administrativeArea": "TX",
      "postalCode": "78757",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": "444-4444-4444"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
   "href": "/pts/v2/payments/7094060020036241803954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7094060020036241803954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/7094060020036241803954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1709406002076"
 },
 "id": "7094060020036241803954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "60616704ST7Q27K2",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2024-03-02T19:00:02Z",
 "tokenInformation": {
  "instrumentidentifierNew": false,
  "instrumentIdentifier": {
   "state": "ACTIVE",
   "id": "7010000000016241111"
  }
  }
}
```

Merchant-Initiated Unscheduled COF Payments with PAN

After the initial CIT unscheduled COF payment, subsequent unscheduled COF transactions are merchant-initiated transactions (MITs).

Prerequisites

The first transaction in an unscheduled COF payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for Authorizing Subsequent MIT Unscheduled COF Payments

These fields are required in a subsequent authorization request for a subsequent unscheduled COF payment:

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. previousTransactionID	• American Express: set to the transaction ID from the original transaction. • Discover: set to the transaction ID from the original transaction. • Visa: set to the last successful transaction ID.
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction.reason	**Set the value to `10`.** **Required only for American Express, Discover and Mastercard.**
processingInformation. authorizationOptions. initiator. storedCredentialUsed	**Set the value to `true`.**
processingInformation. authorizationOptions. initiator. type	**Set the value to `merchant`.**
processingInformation. commerceIndicator	**Set the value to `internet`.**

Card-Specific Required Field for Processing a Merchant-Initiated Transactions

Discover

The listed card requires an additional field:

processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction. originalAuthorizedAmount	**Provide the original transaction amount.**

Country-Specific Required Fields for Authorizing Subsequent Recurring Payments

Include these country-specific required fields for a successful merchant-initiated authorization.

India

These fields are required only with Diners Club in India or with an India-issued card, and you are processing payments through Visa Platform Connect.

installmentInformation.amount

installmentInformation.frequency

installmentInformation.identifier

installmentInformation.paymentType

installmentInformation.sequence

installmentInformation.validationIndicator

Saudi Arabia

These fields are required only if your business is located in Saudi Arabia and you are processing payments through Visa Platform Connect.

authorizationOptions.initiator.merchantInitiatedTransaction.agreementId

recurringPaymentInformation.amountType

REST Example: Authorizing Subsequent MIT Unscheduled COF Payments

Request

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "storedCredentialUsed": "true",
        "type": "merchant",
        "merchantInitiatedTransaction": {
          "previousTransactionId": "123456789619999",
          "originalAuthorizedAmount": "100"   <--Discover Only-->
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "5554327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "79710341A39WTT5W",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

Merchant-Initiated Unscheduled COF Payments with TMS

After the customer-initiated unscheduled COF payment, you can send merchant-initiated unscheduled COF payments using one or more TMS token types:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "customer": {
    "id": "07C9CA98022DA498E063A2598D0AA400"
  }
}
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token.

Including a payment instrument eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"paymentInformation": {
  "paymentInstrument": {
    "id": "07CA24EF20F9E2C9E063A2598D0A8565"
  }
}
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store only a PAN. Including an instrument identifier eliminates the need to include a PAN and the previous transaction's ID.

```
"paymentInformation": {
 "instrumentIdentifier": {
  "id": "7010000000016241111"
 }
}
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Prerequisites

The first transaction in an unscheduled COF payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Supported Card Types

These are the supported card types for processing credentialed transactions:

- American Express
- Mastercard
- Visa

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for MIT Unscheduled COF Payments with TMS

Include these Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.[tokentype].id	Where **[tokentype]** is the TMS token type you are using: • **customer** • **instrumentIdentifier** • **paymentInstrument**

processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction.reason	Set the value to `10`. Required only for American Express, Discover, and Mastercard.
processingInformation.commerceIndicator	Set the value to `internet`.

Instrument Identifier Required Fields

If you are using the **paymentInformation.instrumentIdentifier.id** token, include these required fields in addition to the required fields listed above.

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

Card-Specific Field

The listed card type requires an additional field.

Discover	**processingInformation.authorizationOptions.initiator. merchantInitiatedTransaction.originalAuthorizedAmou** **Provide the original transaction amount.**

Country-Specific Fields

Include these country-specific required fields for a successful merchant-initiated authorization.

India	**These fields are required only with Diners Club in India or with an India-issued card, and you are processing payments through Visa Platform Connect.** **installmentInformation.amount** **installmentInformation.frequency** **installmentInformation.identifier** **installmentInformation.paymentType**

	installmentInformation.sequence installmentInformation.validationIndicator
Saudi Arabia	These fields are required only if your business is located in Saudi Arabia and you are processing payments through Visa Platform Connect. authorizationOptions.initiator.merchantInitiatedTransa recurringPaymentInformation.amountType

Example: MIT Unscheduled COF Payment with TMS Instrument Identifier

Request

```
{
  "processingInformation": {
    "commerceIndicator": "internet"
  },
  "paymentInformation": {
    "card": {
      "expirationMonth": "12",
      "expirationYear": "2031"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "USD"
    },
    "billTo": {
      "firstName": "John",
      "lastName": "Doe",
      "address1": "1 Market St",
      "locality": "san francisco",
      "administrativeArea": "CA",
      "postalCode": "94105",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": "4158880000"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976892714556134003954/reversals"
    },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976892714556134003954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976892714556134003954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697689271513"
 },
 "id": "6976892714556134003954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62699554NNMR6X7R",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T04:21:11Z"
```

```
}
```

Example: MIT Unscheduled COF Payment with TMS Payment Instrument

Request

```
{
 "processingInformation": {
  "commerceIndicator": "internet"
 },
 "paymentInformation": {
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976891300676431103955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976891300676431103955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976891300676431103955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697689130124"
 },
 "id": "6976891300676431103955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
  "paymentSolution": "015"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V0010013022298169667504231315",
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "62699372XNMR85HS",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2023-10-19T04:18:50Z"
}
```

Example: MIT Unscheduled COF Payment with TMS Customer

Request

```
{
 "processingInformation": {
  "commerceIndicator": "internet"
 },
 "paymentInformation": {
  "customer": {
   "id": "080AC9AB60C92AA2E063A2598D0A0C74"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976889582016147703955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6976889582016147703955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976889582016147703955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697688958296"
 },
 "id": "6976889582016147703955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "080AE6DB37B09557E063A2598D0AA4C9"
  },
  "card": {
   "type": "001"
  },
  "customer": {
   "id": "080AC9AB60C92AA2E063A2598D0A0C74"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processingInformation": {
    "paymentSolution": "015"
  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "V0010013022298169667504231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62699842BNN13VA0",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T04:15:58Z"
}
```

Token Management Service Processing

This section provides the information you need in order to process Token Management Service authorization and credit transactions.

> **Important**
>
> Due to mandates from the Reserve Bank of India, Indian merchants cannot store personal account numbers (PANs). Use network tokens instead. For more information on network tokens, see the Network Tokenization section of the *Token Management Service Guide.*

Additional Resources for TMS

For more information, see these guides:

- *Token Management Service Developer Guide*
- *API field reference guide for the REST API*
- Github repositories: *https://github.com/Cybersource*

Authorizing a Payment with a Customer Token

This section provides the information you need to authorize a payment with a customer token.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`

Production in India: POST https://api.in.cybersource.com/pts/v2/payments

Required Fields for Authorizing a Payment with a Customer Token

clientReferenceInformation.code	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.customer.id	**Set to the ID of the customer token you want to use.**

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a Payment with a Customer Token

Request

```
{
 "clientReferenceInformation": {
  "code": "12345678"
 },
  "paymentInformation": {
    "customer": {
      "id": "F45FB3E443AC3C57E053A2598D0A9CFF"
    }

  },
  "orderInformation": {
    "amountDetails": {
      "currency": "USD",
      "totalAmount": "10.00"
    }
  }
}
```

Response to a Successful Request

The request response returns the payment instrument and shipping address IDs that are used as the customer's defaults.

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/7055928871556818104953/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7055928871556818104953"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/7055928871556818104953/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "12345678"
 },
 "id": "7055928871556818104953",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "10.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "shippingAddress": {
   "id": "0F35F0D99AD088B5E063A2598D0AE066"
  },
  "paymentInstrument": {
   "id": "0F35E9CFEA463E34E063A2598D0A3FC2"
  },
  "card": {
   "type": "001"
  },
  "customer": {
   "id": "B21E6717A6F03479E05341588E0A303F"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "67467352CRIISD1G",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2024-01-18T15:48:07Z"
```

```
}
```

REST Example: Authorizing a Payment Using a Customer Token Linked to a Network Token

Request

```
{
 "clientReferenceInformation": {
  "code": "12345678"
 },
 "paymentInformation": {
  "customer": {
   "id": "F60328413BAB09A4E053AF598E0A33DB"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  }
 }
}
```

Response to a Successful Request

The request response returns the payment instrument and shipping address IDs that are used as the customer's defaults.

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6778647071126384904953/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/6778647071126384904953"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6778647071126384904953/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "id": "6778647071126384904953",
 "issuerInformation": {
  "responseRaw": "0110322000000E100002000....."
 },
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "USD"
```

```
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "instrumentIdentifier": {
      "id": "7020000000010603216",
      "state": "ACTIVE"
    },
    "shippingAddress": {
      "id": "F60328413BAE09A4E053AF598E0A33DB"
    },
    "paymentInstrument": {
      "id": "F6032841BE33098EE053AF598E0AB0A5"
    },
    "card": {
      "type": "002"
    },
    "customer": {
      "id": "F60328413BAB09A4E053AF598E0A33DB"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "08244117"
  },  "processingInformation": {    "paymentSolution": "014"  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "50015OU4U5UYXLV127XTONYN49CL1",
    "merchantNumber": "000844028303882",
    "approvalCode": "831000",
    "networkTransactionId": "0602MCC603474",
    "transactionId": "0602MCC603474",
    "responseCode": "00",
    "avs": {
      "code": "Y",
      "codeRaw": "Y"
    }
  },
  "reconciliationId": "EUHW1EMHIZ3O",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-03-03T17:31:48Z"
}
```

Authorizing a Payment with a Non-Default Shipping Address

This section provides the information you need in order to make a payment with a non-default shipping address.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`
Production in India: `POST https://api.in.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Payment with a Non-Default Shipping Address

clientReferenceInformation.code	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.customer.id	**Set to the ID of the customer token you want to use.**
paymentInformation.shippingAddress.id	**Set to the ID of the shipping address token you want to use.**

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a Payment with a Non-Default Shipping Address

Request

```
{
 "clientReferenceInformation": {
  "code": "12345678"
 },
  "paymentInformation": {
    "customer": {
      "id": "F45FB3E443AC3C57E053A2598D0A9CFF"
    },
    "shippingAddress": {
      "id": "F45FD8DE51B99E9CE053A2598D0AFDFA"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "currency": "USD",
      "totalAmount": "10.00"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/7055949037316786904953/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7055949037316786904953"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/7055949037316786904953/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "12345678"
 },
 "id": "7055949037316786904953",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "10.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7030000000014831523",
   "state": "ACTIVE"
  },
  "shippingAddress": {
   "id": "F45FD8DE51B99E9CE053A2598D0AFDFA"
  },
  "paymentInstrument": {
   "id": "F45FE45E7993C7DBE053A2598D0AED19"
  },
  "card": {
   "type": "001"
  },
  "customer": {
   "id": "F45FB3E443AC3C57E053A2598D0A9CFF"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "674679208RIKQ52K",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2024-01-18T16:21:44Z"
}
```

Authorizing a Payment with a Non-Default Payment Instrument

This section provides the information you need in order to authorize a payment with a non-default payment instrument.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`
Production in India: `POST https://api.in.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Payment with a Non-Default Payment Instrument

clientReferenceInformation.code

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

paymentInformation.paymentInstrument.id **Set to the ID of the payment instrument token you want to use.**

Related Information

- *API field reference guide for the REST API*

Optional Fields for Authorizing a Payment with a Non-Default Payment Instrument

You can use these optional fields to include additional information when authorizing a payment with a non-default payment instrument.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

paymentInformation.card.type

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a Payment with a Non-Default Payment Instrument

Request

```
{
 "clientReferenceInformation": {
  "code": "12345678"
 },
  "paymentInformation": {
    "paymentInstrument": {
      "id": "0F3BB131F8143A58E063A2598D0AB921"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "currency": "USD",
      "totalAmount": "10.00"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/7055952648586653304951/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7055952648586653304951"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/7055952648586653304951/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "12345678"
 },
 "id": "7055952648586653304951",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "10.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "paymentInstrument": {
   "id": "0F3BB131F8143A58E063A2598D0AB921"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "67468244CRIL0U0Y",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2024-01-18T16:27:45Z"
}
```

Authorizing a Payment with a Payment Instrument

This section provides the information you need in order to authorize a payment with a payment instrument.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`
Production in India: `POST https://api.in.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Payment with a Payment Instrument

clientReferenceInformation.code	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.paymentInstrument.id	**Set to the ID of the payment instrument token you want to use.**

Related Information

- *API field reference guide for the REST API*

Optional Fields for Authorizing a Payment with a Payment Instrument

You can use these optional fields to include additional information when authorizing a payment with a payment instrument.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

paymentInformation.card.type

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a Payment with a Payment Instrument

Request

```
{
 "clientReferenceInformation": {
  "code": "12345678"
 },
  "paymentInformation": {
    "paymentInstrument": {
      "id": "F4D5E715F7BD9910E053A2598D0A7278"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "currency": "USD",
      "totalAmount": "10.00"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6765713628736138103955/reversals"
```

```
        },
        "self": {
            "method": "GET",
            "href": "/pts/v2/payments/6765713628736138103955"
        },
        "capture": {
            "method": "POST",
            "href": "/pts/v2/payments/6765713628736138103955/captures"
        }
    },
    "clientReferenceInformation": {
        "code": "12345678"
    },
    "id": "6765713628736138103955",
    "orderInformation": {
        "amountDetails": {
            "authorizedAmount": "10.00",
            "currency": "USD"
        }
    },
    "paymentAccountInformation": {
        "card": {
            "type": "001"
        }
    },
    "paymentInformation": {
        "tokenizedCard": {
            "type": "001"
        },
        "instrumentIdentifier": {
            "id": "7010000000016241111",
            "state": "ACTIVE"
        },
        "paymentInstrument": {
            "id": "F4D5E715F7BD9910E053A2598D0A7278"
        },
        "card": {
            "type": "001"
        },
        "customer": {
            "id": "F4D5E715F75E9910E053A2598D0A7278"
        }
    },
    "pointOfSaleInformation": {
        "terminalId": "111111"
    },
    "processorInformation": {
        "approvalCode": "888888",
        "networkTransactionId": "123456789619999",
        "transactionId": "123456789619999",
        "responseCode": "100",
        "avs": {
            "code": "X",
            "codeRaw": "I1"
        }
    },
    "reconciliationId": "60561224BE37KN5W",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2023-02-16T18:16:03Z"
 }
```

Authorize a Payment with an Instrument Identifier

This section provides the information you need in order to authorize a payment with an instrument identifier token.

Endpoint

Production: `POST https://api.cybersource.com/pts/v2/payments`
Test: `POST https://apitest.cybersource.com/pts/v2/payments`
Production in India: `POST https://api.in.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Payment with an Instrument Identifier

clientReferenceInformation.code

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

paymentInformation.instrumentIdentifier.id **Set to the ID of the instrument identifier token you want to use.**

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a Payment with an Instrument Identifier

Request

```
{
  "clientReferenceInformation": {
    "code": "12345678"
  },
    "paymentInformation": {
      "instrumentIdentifier": {
        "id": "7010000000016241111"
      }
    },
    "orderInformation": {
      "amountDetails": {
        "currency": "USD",
        "totalAmount": "10.00"
      }
    }
}
```

Response to a Successful Request

```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/7055955288186053404953/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7055955288186053404953"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/7055955288186053404953/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "12345678"
 },
 "id": "7055955288186053404953",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "10.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "instrumentIdentifier": {
   "id": "7010000000016241111",
   "state": "ACTIVE"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "1"
  }
 },
 "reconciliationId": "67468271CRIL0U24",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2024-01-18T16:32:09Z"
}
```

REST Example: Authorizing a Payment with an Instrument Identifier While Creating TMS Tokens

Request

```
{
 "clientReferenceInformation": {
  "code": "TC50171_3"
 },
 "processingInformation": {
  "actionList": [
   "TOKEN_CREATE"
  ],
  "actionTokenTypes": [
   "customer",
   "paymentInstrument",
   "shippingAddress"
  ]
 },
 "paymentInformation": {
  "instrumentIdentifier": {
   "id": "7010000000016241111"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "USD"
  },
  "billTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
   "administrativeArea": "CA",
   "postalCode": "94105",
   "country": "US",
   "email": "test@cybs.com",
   "phoneNumber": "4158880000"
  },
  "shipTo": {
   "firstName": "John",
   "lastName": "Doe",
   "address1": "1 Market St",
   "locality": "san francisco",
      "administrativeArea": "CA",
      "postalCode": "94105",
      "country": "US"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/7114679840376687203955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/7114679840376687203955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/7114679840376687203955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "id": "7114679840376687203955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111",
      "state": "ACTIVE"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "623971212U7PN4IU",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2024-03-26T15:46:24Z",
  "tokenInformation": {
    "shippingAddress": {
      "id": "14930C904FC4D97BE063A2598D0AE0F1"
    },
    "paymentInstrument": {
      "id": "149310A4A924E911E063A2598D0A47AD"
    },
    "customer": {
      "id": "14930C904FC1D97BE063A2598D0AE0F1"
    }
  }
}
```

Authorize a Payment While Ignoring Network Token

This section describes how to authorize a payment ignoring a network token.

Endpoint

Test: `POST https://apitest.cybersource.com/pts/v2/payments`
Production: `POST https://api.cybersource.com/pts/v2/payments`
Production in India: `POST https://api.in.cybersource.com/pts/v2/payments`

Required Fields for Authorizing a Payment While Ignoring Network Token

clientReferenceInformation.code

paymentInformation.customer.id

paymentInformation.paymentInformation.id

paymentInformation.shippingAddress.id

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

processingInformation.capture

processingInformation.commerceIndicator	
tokenInformation.networkTokenOption	**Set value to `ignore`.**

Related Information

- *API Field Reference for the REST API*

REST Example: Authorizing a Payment While Ignoring Network Token

Request

```
{
   "clientReferenceInformation": {
      "code": "RTS-Auth"
   },
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "expirationMonth": "12",
         "type": "001"
      },
      "instrumentIdentifier": {
         "id": "7010000000016241111"
      }
   },
   "orderInformation": {
      "amountDetails": {
         "currency": "USD",
         "totalAmount": "1.00"
      }
   },
   "processingInformation": {
      "capture": "false",
      "commerceIndicator": "internet"
   },
   "tokenInformation": {
      "networkTokenOption": "ignore"
   }
}
```

Response to a Successful Request

```
{
   "_links": {
      "authReversal": {
         "method": "POST",
         "href": "/pts/v2/payments/6769913443166412604951/reversals"
      },
      "self": {
         "method": "GET",
         "href": "/pts/v2/payments/6769913443166412604951"
      },
      "capture": {
         "method": "POST",
      "href": "/pts/v2/payments/6769913443166412604951/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "RTS-Auth"
  },
  "id": "6769913443166412604951",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "1.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7030000000014911515",
      "state": "ACTIVE"
    },
    "shippingAddress": {
      "id": "F537CE8DBA2F032CE053AF598E0A64F2"
    },
    "paymentInstrument": {
      "id": "F537E3D12322416EE053AF598E0AD771"
    },
    "card": {
      "type": "001"
    },
    "customer": {
      "id": "F537CE8DBA2C032CE053AF598E0A64F2"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "V0010013019326121174070050420",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "744295942E2LY3F8",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-02-21T14:55:44Z"
```

```
}
```

Authorizing a Payment with a Legacy Token

This section describes how to authorize a payment with a legacy token.

Endpoint

Production: POST https://api.cybersource.com/pts/v2/payments
Test: POST https://apitest.cybersource.com/pts/v2/payments

Required Fields for Authorizing a Payment with a Legacy Token

clientReferenceInformation.code	
paymentInformation.legacyToken.id	**Include the ID of the legacy token you want to use to authorize a payment.**
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	

Related Information

- *API field reference guide for the REST API*

REST Example: Authorizing a Payment with a Legacy Token

Request

```
{
  "clientReferenceInformation": {
    "code": "12345678"
  },
  "paymentInformation": {
    "legacyToken": {
      "id": "B21E6717A6F03479E05341588E0A303F"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "22.00",
      "currency": "USD"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
```

```
   "method": "POST",
   "href": "/pts/v2/payments/7055956342476789004951/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7055956342476789004951"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/7055956342476789004951/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "12345678"
 },
 "id": "7055956342476789004951",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "22.00",
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "approvalCode": "888888",
  "networkTransactionId": "123456789619999",
  "transactionId": "123456789619999",
  "responseCode": "100",
  "avs": {
   "code": "X",
   "codeRaw": "I1"
  }
 },
 "reconciliationId": "67468431FRIIS246",
 "status": "AUTHORIZED",
 "submitTimeUtc": "2024-01-18T16:33:54Z"
}
```

Making a Credit with a Customer Token

This section describes how to make a credit with a customer token.

Endpoint

Test: POST https://apitest.cybersource.com/pts/v2/credits
Production: POST https://api.cybersource.com/pts/v2/credits
Production in India: POST https://api.in.cybersource.com/pts/v2/credits

Required Fields for Making a Credit with a Customer Token

clientReferenceInformation.code	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.customer.id	**Set to the ID of the customer token you want to use.**

Related Information

- *API field reference guide for the REST API*

REST Example: Making a Credit with a Customer Token

Request

```
{
 "clientReferenceInformation": {
  "code": "12345678"
 },
  "paymentInformation": {
    "customer": {
      "id": "F45FB3E443AC3C57E053A2598D0A9CFF"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "currency": "USD",
      "totalAmount": "10.00"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
      "href": "/pts/v2/credits/7055967677826132904951/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/credits/7055967677826132904951"
    }
  },
  "clientReferenceInformation": {
    "code": "12345678"
  },
  "creditAmountDetails": {
    "currency": "USD",
    "creditAmount": "10.00"
  },
  "id": "7055967677826132904951",
  "orderInformation": {
    "amountDetails": {
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7030000000014831523",
      "state": "ACTIVE"
    },
    "shippingAddress": {
      "id": "F45FD8DE51B99E9CE053A2598D0AFDFA"
    },
    "paymentInstrument": {
      "id": "F45FE45E7993C7DBE053A2598D0AED19"
    },
    "card": {
      "type": "001"
    },
    "customer": {
      "id": "F45FB3E443AC3C57E053A2598D0A9CFF"
    }
  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "V0010013019326121538313096266",
    "approvalCode": "888888",
    "responseCode": "100"
  },
  "reconciliationId": "67444961BRIL0BB8",
  "status": "PENDING",
  "submitTimeUtc": "2024-01-18T16:52:48Z"
}
```

Making a Credit with a Non-Default Payment Instrument

This section describes how to make a credit with a non-default payment instrument.

Endpoint

Test: POST https://apitest.cybersource.com/pts/v2/credits
Production: POST https://api.cybersource.compts/v2/credits
Production in India: POST https://api.in.cybersource.compts/v2/credits

Required Fields for Making a Credit with a Non-Default Payment Instrument

clientReferenceInformation.code	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.paymentInstrument.id	**Set to the ID of the payment instrument token that you want to use.**

Related Information

- *API field reference guide for the REST API*

Optional Fields for Making a Credit with a Non-Default Payment Instrument

You can use these optional fields to include additional information when making a credit with a non-default payment instrument.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

paymentInformation.card.type

Related Information

- *API field reference guide for the REST API*

REST Example: Making a Credit with a Non-Default Payment Instrument

Request

```
{
 "clientReferenceInformation": {
  "code": "12345678"
 },
  "paymentInformation": {
    "paymentInstrument": {
      "id": "0F3BB131F8143A58E063A2598D0AB921"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "currency": "USD",
      "totalAmount": "10.00"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
   "href": "/pts/v2/credits/7055968581386446104953/voids"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/credits/7055968581386446104953"
  }
 },
 "clientReferenceInformation": {
  "code": "12345678"
 },
 "creditAmountDetails": {
  "currency": "USD",
  "creditAmount": "10.00"
  },
  "id": "7055968581386446104953",
  "orderInformation": {
    "amountDetails": {
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111",
      "state": "ACTIVE"
    },
    "paymentInstrument": {
      "id": "0F3BB131F8143A58E063A2598D0AB921"
    },
    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "approvalCode": "888888",
    "responseCode": "100"
  },
  "reconciliationId": "67445196PRILCQCN",
  "status": "PENDING",
  "submitTimeUtc": "2024-01-18T16:54:18Z"
}
```

Making a Credit with a Payment Instrument

This section describes how to make a credit with a payment instrument.

Endpoint

Test: POST https://apitest.cybersource.com/pts/v2/credits
Production: POST https://api.cybersource.compts/v2/credits
Production in India: POST https://api.in.cybersource.compts/v2/credits

Required Fields for Making a Credit with a Payment Instrument

clientReferenceInformation.code

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount	
paymentInformation.paymentInstrument.id	**Set to the ID of the payment instrument token you want to use.**

Related Information

- *API field reference guide for the REST API*

REST Example: Making a Credit with a Payment Instrument

Request

```
{
 "clientReferenceInformation": {
  "code": "12345678"
 },
  "paymentInformation": {
    "paymentInstrument": {
      "id": "F4D5E715F7BD9910E053A2598D0A7278"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "currency": "USD",
      "totalAmount": "10.00"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
   "href": "/pts/v2/credits/7055969586686467104953/voids"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/credits/7055969586686467104953"
  }
 },
 "clientReferenceInformation": {
  "code": "12345678"
 },
 "creditAmountDetails": {
  "currency": "USD",
  "creditAmount": "10.00"
 },
 "id": "7055969586686467104953",
 "orderInformation": {
  "amountDetails": {
   "currency": "USD"
  }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111",
      "state": "ACTIVE"
    },
    "paymentInstrument": {
      "id": "F4D5E715F7BD9910E053A2598D0A7278"
    },
    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "approvalCode": "888888",
    "responseCode": "100"
  },
  "reconciliationId": "67446174JRIKXXHB",
  "status": "PENDING",
  "submitTimeUtc": "2024-01-18T16:55:59Z"
}
```

Making a Credit with an Instrument Identifier

This section describes how to make a credit with an instrument identifier token.

Endpoint

Test: `POST https://apitest.cybersource.com/pts/v2/credits`
Production: `POST https://api.cybersource.compts/v2/credits`
Production in India: `POST https://api.in.cybersource.compts/v2/credits`

Required Fields for Making a Credit with an Instrument Identifier

clientReferenceInformation.code

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

paymentInformation.paymentInstrument.id **Set to the ID of the payment instrument token you want to use.**

Related Information

- *API field reference guide for the REST API*

REST Example: Making a Credit with an Instrument Identifier

Request

```
{
 "clientReferenceInformation": {
  "code": "12345678"
 },
  "paymentInformation": {
    "instrumentIdentifier": {
      "id": "7010000000016241111"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "currency": "USD",
      "totalAmount": "10.00"
    }
  }
}
```

Response to a Successful Request

```
{
 "_links": {
  "void": {
   "method": "POST",
   "href": "/pts/v2/credits/7055970261066212404951/voids"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/credits/7055970261066212404951"
  }
 },
 "clientReferenceInformation": {
  "code": "12345678"
 },
 "creditAmountDetails": {
  "currency": "USD",
  "creditAmount": "10.00"
 },
 "id": "7055970261066212404951",
 "orderInformation": {
  "amountDetails": {
   "currency": "USD"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "7010000000016241111",
      "state": "ACTIVE"
    },
    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "approvalCode": "888888",
    "responseCode": "100"
  },
  "reconciliationId": "67445198PRILCQCQ",
  "status": "PENDING",
  "submitTimeUtc": "2024-01-18T16:57:06Z"
}
```

Making a Credit with a Legacy Token

This section describes how to make a credit with a legacy token.

Endpoint

Test: `POST https://apitest.cybersource.com/pts/v2/credits`
Production: `POST https://api.cybersource.com/pts/v2/credits`
Production in India: `POST https://api.in.cybersource.com/pts/v2/credits`

Required Fields for Making a Credit with a Legacy Token

clientReferenceInformation.code	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.legacyToken.id	**Include the ID of the legacy token that you want to use to authorize a payment.**

Related Information

- *API field reference guide for the REST API*

REST Example: Making a Credit with a Legacy Token

Request

```
{
  "clientReferenceInformation": {
    "code": "12345678"
  },
  "paymentInformation": {
    "legacyToken": {
      "id": "B21E6717A6F03479E05341588E0A303F"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "22.00",
      "currency": "USD"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/credits/7055970562096509704953/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/credits/7055970562096509704953"
    }
  },
  "clientReferenceInformation": {
    "code": "12345678"
  },
  "creditAmountDetails": {
    "currency": "USD",
    "creditAmount": "22.00"
  },
  "id": "7055970562096509704953",
  "orderInformation": {
    "amountDetails": {
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "approvalCode": "888888",
    "responseCode": "100"
  },
  "reconciliationId": "67444779FRILJT84",
  "status": "PENDING",
  "submitTimeUtc": "2024-01-18T16:57:36Z"
}
```