Credentialed Transactions

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 Unpublished rights reserved under the copyright laws of the United States.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

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

Version: 26.02.01

Contents

Credentialed Transactions Developer Guide

This section describes how to use this developer 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 using credentials 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 credit card services into your existing order management system.
Visit the *Cybersource documentation hub* to find additional processor-specific versions of this guide and additional technical documentation.

Convention

This statement appears in this document:

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

Customer Support

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

Recent Revisions to This Document

26.02.01

Added industry and use case information to the introduction topics. See *Industry Practice Transactions* on page 11 and *Standing Instruction Transactions* on page 12.

25.12.01

This revision contains only editorial changes and no technical updates.

25.11.01

Removed Mastercard required field for retrieving customer credentials during a CIT request. See *Using Stored Customer Credentials During a CIT* on page 25.

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 Credentialed Transactions

Credentialed transactions, also known as credentials#on#file (COF) or card#on#file transactions, are payments that either store a customer's payment credentials for future use or use previously stored credentials to complete a transaction. All COF transactions begin with a customer-initiated transaction, in which the customer actively participates, such as a card#present purchase, online checkout, or use of a stored credential.

Benefits of Credentialed Transactions

Merchants following the stored credentials framework experience these benefits:

- Better visibility into transaction risk.
- Improved authorization success rates.
- A smoother customer experience.
- Fewer disputes and customer complaints.
- Use of Real Time Visa Account Updater for fresher card details.

For more information on the stored credentials framework, see *Improving Authorization Management for Transactions with Stored Credentials*.

Types of Credentialed Transactions

There are several types of credentialed transactions:

- Customer-initiated transaction (CIT): During a CIT, customers can elect to have their credentials stored for future CITs or for merchant#initiated transactions (MITs).
- Merchant-initiated transaction (MIT): A MIT is processed without the customer's active involvement and include these transactions:
  - Industry practice transaction: This MIT is performed as a subsequent transaction 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: This MIT is performed to follow agreed-upon instructions from the customer for the provision of goods and services.

Industry Practice Transactions

Industry practice transactions are MITs performed as follow#on actions to a previous CIT. Although not all of them require stored credentials, repeated use of credentials qualifies them as COF transactions.
These industry practice transactions and industry examples are available with your processor:

- *Delayed charges*: Used to add charges after the initial transaction is complete. Examples: hotels (minibar, damages), car rentals (tolls), travel (post-trip charges), and health and wellness add-ons.
- *Incremental charges*: Used when an amount exceeds the original authorization. Examples: extending hotel stays, adding rental car insurance, restaurant gratuities, and event upgrades.
- *Reauthorizations*: Used when an authorization expires before fulfillment. Examples: long hotel stays, extended rental agreements, multi-week equipment rentals, and delayed subscription boxes.
- *Resubmissions*: Used when a previous authorization attempt fails. Examples: utility auto-pay retries, telecom billing, insurance premiums, and online membership renewals.
- *No-shows*: Used when a customer fails to appear for a reserved service for these industries: hotels, rentals, healthcare missed appointments, and restaurant reservation deposits.

Business Center Transactions

You can create an industry practice transaction in the Business Center by requesting a new authorization. Go to the Transaction Management section and confirm that the new authorization is a MIT. Choose one of these reason types for the authorization:

- Account Top Up
- Delayed Charges
- No Show
- Reauthorization
- Resubmission

This process requires you to have already stored the customer's credentials from a previous customer-initiated transaction. For more information on storing a customer's credentials in the Business Center, see *Customer-Initiated Transactions with Credentials on File* on page 17.
To create an incremental transaction in the Business Center, choose one of these options:

- Account Top Up
- No Show

Standing Instruction Transactions

Standing instruction transactions are MITs that rely on stored credentials and follow agreed#upon customer instructions for scheduled or ongoing payments. These transactions must comply with the stored credentials framework, which ensures secure storage and use of customer payment data. All standing instruction transactions begin with a CIT, when customers elect to store their credentials.
These standing instruction transactions and industry examples are available with your processor:

- *Unscheduled COF*: Occasional, non#scheduled charges that are made under a customer authorization for these industries:
  - Rideshare and transportation: cleaning fees, damage fees
  - Home services: irregular invoice-based jobs, such as repairs
  - Professional services: unplanned billable hours or fees
  - E#commerce: back-order fulfillment outside a schedule
- *Installments*: A fixed purchase that is split into multiple scheduled payments for these industries:
  - Retail and electronics: installment plans for device purchases
  - Furniture and home goods: multi#month payment plans
  - Education: tuition installment schedules
  - Healthcare financing: payment plans for procedures
- *Recurring*: Repeated charges for ongoing services for these industries:
  - Streaming services: video, music, gaming subscriptions
  - Fitness and wellness: gym memberships, coaching subscriptions
  - Insurance: monthly premiums
  - Software and SaaS: business application licenses
- *Subscription Transactions for Mastercard*: Mastercard#specific recurring billing for subscription#based services for these industries:
  - Digital media: news, magazines, premium content
  - Subscription boxes: food kits, beauty boxes, hobby crates
  - Online services: cloud storage, identity monitoring
  - Educational platforms: e#learning subscriptions
- *Standing Order Transactions for Mastercard*: Merchant#initiated charges made at regular, agreed-upon intervals for these industries:
  - Utilities: monthly electricity, water, gas payments
  - Telecommunications: phone and internet service billing
  - Loan and mortgage payments: fixed monthly obligations
  - Charitable donations: recurring monthly contributions

Requirements for Standing Instruction Transactions

Merchants who offer stored credentials must:

- Disclose to cardholders how their credentials will be used.
- Obtain the customer's consent to store their credentials.
- Notify customers when the terms of use change.
- Inform the card issuer during an authorization that the credentials are stored on file.
- Identify all transactions that use stored credentials.

Recurring Billing for Recurring Payments

If you are using the Recurring Billing service, do not use this document. Cybersource saves and stores payment credentials for recurring transactions, ensuring compliance with COF best practices.
For more information on Recurring Billing, see *Recurring Billing*.

Transaction-Specific Fields

To make an authorization request into a credentialed transaction, you must include additional fields that inform Cybersource to either store the customer's payment information for future use, or to use an already stored card-on-file for the payment. This section describes the additional required fields that create an initial and subsequent credentialed transaction.

Initial Transactions

For an initial transaction, include these fields with a standard authorization request:

processingInformation.authorizationOptions. initiator. credentialStoredOnFile	Set the value to `true`.
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction.reason	Some processors and card types require a reason code when storing payment credentials.
processingInformation. authorizationOptions. initiator. type	Set the value to `customer`.
processingInformation. commerceIndicator	Set to one of these possible values: • `internet`: Online transaction. • `MOTO`: Mail order/telephone order transaction. • A payer authentication value. See *Payer Authentication Values* on page 201.

```
{
```

```
  "processingInformation": {
   "commerceIndicator": "internet",
   "authorizationOptions": {
    "initiator": {
     "type": "customer",
     "credentialStoredOnFile": true,
     "merchantInitiatedTransaction": {
      "reason": "7"
     }
    }
   }
  }
}
```

When you receive the initial transaction response, save the transaction identifier, which is located in the **id** field. You need the transaction identifier for subsequent transactions. If you are using the Token Management Service (TMS), Cybersource stores the transaction identifier for you.

This table shows the fields required for each type of CIT and initial transaction.

		type	commerceIndicator	reason (Mastercard)
	CIT Transaction without Storing Credential	✕	any indicator	✕
	CIT Transaction while Storing Credential	customer	internet payer auth	✕
Standing Instructions	Initial Installment Transaction	customer	internet payer auth	9
Standing Instructions	Initial Recurring Transaction	customer	internet payer auth	✕
Standing Instructions	Initial Subscription Transaction (Mastercard)	customer	internet payer auth	7
Standing Instructions	Initial Standing Order Transaction (Mastercard)	customer	internet payer auth	8

Subsequent Transactions

For a subsequent transaction, include these fields with a standard authorization request:

processingInformation.authorizationOptions.. initiator. merchantInitiatedTransaction. previousTransactionID	American Express: Set the value to the transaction ID from the original transaction. • Discover: Set the value to the transaction ID from the original transaction. • Visa: set the value to the last successful transaction ID.
processingInformation. authorizationOptions. initiator. merchantInitiatedTransaction.reason	**Some processors and card types require a reason code when you use stored payment credentials.**
processingInformation. authorizationOptions. initiator. storedCredentialUsed	**Set the value to `true`.**
processingInformation. authorizationOptions. initiator. type	**Set the value to `merchant` for MIT transactions.**
processingInformation. commerceIndicator	**Set to one of these possible values:** • `install`: Installment payment • `internet`: E-commerce order • `MOTO`: Mail order or telephone order • `recurring`: Recurring payment • A payer authentication value. See *Payer Authentication Values* on page 201.

```
{
 "processingInformation": {
  "commerceIndicator": "internet",
  "authorizationOptions": {
   "initiator": {
    "type": "merchant",
    "storedCredentialUsed": true,
    "merchantInitiatedTransaction": {
     "reason": "7",
     "previousTransactionId": "123456789123"
    }
   }
  }
 }
}
```

This table shows the values for subsequent authorization fields.

	previous TransactionID	commerce Indicator	type	reason	stored CredentialU
Delayed	✗	internet	merch	2	✗
Incremental	✓	internet	merch	5	✗
No Show	✓	internet	merch	4	✗
Partial-Shipment	✓	internet	merch	6	✗
Reauthorization	✓	internet	merch	3	✗
Resubmission	✓	internet	merch	1	✗
CIT Ad Hoc (COF)	✓	internet vbv	cust	✗	✓
Unsched MIT (UCOF)	✓	internet	merch	✗	✓
Installment	✓	install	merch	9	✓
Recurring	✓	recurring	merch	7	✓
Subscription	✓	recurring	merch	7	✓
Standing Order	✓	recurring	merch	8	✓

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 83.

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

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 202.

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": "ABC"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/652818719894607630304"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
  },
  "clientReferenceInformation": {
```

```
      "code": "1652818719876"
    },
    "id": "6528187198946076303004",
    "orderInformation": {
      "amountDetails": {
        "authorizedAmount": "100.00",
        "currency": "ABC"
      }
    },
    "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 202.

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": "ABC"
  },
  "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": "ABC"
  }
 },
 "paymentAccountInformation": {
  "card": {
   "type": "001"
  }
 },
 "paymentInformation": {
  "tokenizedCard": {
   "type": "001"
  },
  "card": {
   "type": "001"
  }
 },
 "pointOfSaleInformation": {
  "terminalId": "111111"
 },
 "processorInformation": {
  "paymentAccountReferenceNumber": "V001001302229816966750 4231315",
  "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"
   }
  }
}
```


Using Stored Customer Credentials During a CIT

After customers store their credentials on file, you can retrieve these credentials to use with subsequent transactions when the customer is present.

Endpoint

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

Required Fields for Using Customer Credentials During a CIT

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 202.

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 the value to `true`.**

Card-Specific Required Field for Retrieving Customer Credentials During a MIT

Discover

Discover requires the authorization amount from the original transaction in addition to the above required fields.

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction. originalAuthorizedAmount

REST Example: Using 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": "ABC",
      "originalAmount": "100"
```

```
            // Discover card Only
        }
    },
    "paymentInformation": {
        "card": {
            "expirationYear": "2031",
            "number": "4111xxxxxxxxxxx",
            "expirationMonth": "12"
        }
    },
   "processorInformation": {
      "transactionId": "123456789610000"
   }
}
```

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 28
- *Merchant-Initiated Delayed Transaction with TMS* on page 32

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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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 202.

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.	• American Express: set to the transaction ID from the original transaction.

merchantInitiatedTransaction. previousTransactionId	• 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": "ABC"
     }
```

```
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxx",
      "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/653421365351659900301"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6534213653516599003001/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653421365327"
  },
  "id": "6534213653516599003001",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "120.00",
      "currency": "ABC"
    }
  },
  "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 202.

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 Required 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.**

REST 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": "ABC"
   },
   "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/697692283045693400 3954"
   },
   "capture": {
    "method": "POST",
    "href": "/pts/v2/payments/6976922830456934003954/captures"
   }
  },
  "clientReferenceInformation": {
   "code": "1697692283160"
  },
  "id": "6976922830456934003954",
  "orderInformation": {
   "amountDetails": {
    "authorizedAmount": "102.21",
    "currency": "ABC"
```

```
    }
   },
   "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"
}
```


REST 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": "ABC"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976917718796256603955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/697691771879625660 3955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976917718796256603955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697691771976"
  },
  "id": "6976917718796256603955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "ABC"
    }
  },
  "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": "V001001302229816966750 4231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62700629BNN13VGW",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:02:52Z"
}
```


REST 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": "ABC"
    }
  }
}
```

Response to a Successful Request


```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976916433716228003955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/697691643371622800 3955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976916433716228003955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697691643458"
 },
 "id": "6976916433716228003955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "ABC"
  }
 },
 "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

- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

Limitations

You can request up to 100 incremental authorizations for each transaction, in addition to the original authorization.
Interchange optimization and split shipments are not supported.

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": "ABC"
      }
   },
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "number": "4111xxxxxxxxxxx",
         "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/653322500655686000 3002"
     },
     "capture": {
       "method": "POST",
       "href": "/pts/v2/payments/6533225006556860003002/captures"
     }
   },
   "clientReferenceInformation": {
     "code": "1653322500637"
   },
   "id": "6533225006556860003002",
   "orderInformation": {
     "amountDetails": {
       "authorizedAmount": "100.00",
       "currency": "ABC"
     }
   },
   "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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB

- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

Limitations

You can request up to 100 incremental authorizations for each transaction, in addition to the original authorization.
Interchange optimization and split shipments are not supported.

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 Required 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.**

REST 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": "ABC"
  },
  "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": "ABC"
    }
   },
   "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"
}
```

REST 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": "ABC"
  }
 }
}
```


Response to a Successful Request


```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/697691771879625660 3955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976917718796256603955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697691771976"
 },
 "id": "6976917718796256603955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "ABC"
  }
 },
```

```
  "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"
}
```


REST 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": "ABC"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/697691643371622800 3955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697691643458"
  },
  "id": "697691643371622800 3955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "ABC"
    }
  },
  "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 56
- *Merchant-Initiated Reauthorization Transactions with TMS* on page 60

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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta

- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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

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 202.

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.	• American Express: set to the transaction ID from the original transaction.

merchantInitiatedTransaction. previousTransactionId	• 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. merchantInitiatedTransaction. originalAuthorizedAmount	**Provide the original transaction amount.**

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": "ABC"
      }
    },
    "paymentInformation": {
      "card": {
        "expirationYear": "2031",
        "number": "4111xxxxxxxxxxx",
        "expirationMonth": "12"
      }
    }
}
```

Response to a Successful Request

```
{
    "_links": {
      "authReversal": {
        "method": "POST",
        "href": "/pts/v2/payments/6541178668686490403003/reversals"
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/654117866868649040303"
      },
      "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/6541178668686490403003/captures"
      }
    },
    "clientReferenceInformation": {
      "code": "1654117866849"
    },
    "id": "654117866868649040303",
    "orderInformation": {
      "amountDetails": {
        "authorizedAmount": "130.00",
        "currency": "ABC"
      }
    },
    "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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)

- Mastercard
- Visa
- Visa Electron

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 Required 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.**

REST 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": "ABC"
   },
   "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/697692283045693400 3954"
   },
   "capture": {
    "method": "POST",
    "href": "/pts/v2/payments/6976922830456934003954/captures"
   }
  },
  "clientReferenceInformation": {
   "code": "1697692283160"
  },
  "id": "6976922830456934003954",
  "orderInformation": {
   "amountDetails": {
    "authorizedAmount": "102.21",
    "currency": "ABC"
```

```
    }
   },
   "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"
}
```


REST 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": "ABC"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976917718796256603955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/697691771879625660 3955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976917718796256603955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697691771976"
  },
  "id": "6976917718796256603955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "ABC"
    }
  },
  "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": "V001001302229816966750 4231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62700629BNN13VGW",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:02:52Z"
}
```


REST 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": "ABC"
    }
  }
}
```

Response to a Successful Request


```
{
 "_links": {
  "authReversal": {
   "method": "POST",
   "href": "/pts/v2/payments/6976916433716228003955/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/697691643371622800 3955"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/6976916433716228003955/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1697691643458"
 },
 "id": "6976916433716228003955",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "ABC"
  }
 },
 "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": "V001001302229816966750 4231315",
    "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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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

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": "ABC"
      }
   },
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "number": "4111xxxxxxxxxxx",
         "expirationMonth": "12"
      }
   }
}
```

Response to a Successful Request

```
{
   "_links": {
      "authReversal": {
```

```
        "method": "POST",
        "href": "/pts/v2/payments/6534232293716260503006/reversals"
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/653423229371626050306"
      },
      "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/6534232293716260503006/captures"
      }
    },
    "clientReferenceInformation": {
      "code": "1653423229353"
    },
    "id": "6534232293716260503006",
    "orderInformation": {
      "amountDetails": {
        "authorizedAmount": "100.00",
        "currency": "ABC"
      }
    },
    "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": "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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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

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 Required 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.**

REST 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": "ABC"
  },
  "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/697692283045693400 3954"
   },
   "capture": {
    "method": "POST",
    "href": "/pts/v2/payments/6976922830456934003954/captures"
   }
  },
  "clientReferenceInformation": {
   "code": "1697692283160"
  },
  "id": "6976922830456934003954",
  "orderInformation": {
   "amountDetails": {
    "authorizedAmount": "102.21",
    "currency": "ABC"
   }
  },
  "paymentAccountInformation": {
   "card": {
    "type": "001"
   }
  },
  "paymentInformation": {
   "tokenizedCard": {
    "type": "001"
   },
   "instrumentIdentifier": {
    "id": "7010000000016241111",
    "state": "ACTIVE"
   },
   "card": {
    "type": "001"
   }
  },
  "pointOfSaleInformation": {
   "terminalId": "111111"
```

```
  },
  "processingInformation": {
    "paymentSolution": "015"
  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "V001001302229816966750 4231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62700184NNMR6XFK",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:11:23Z"
}
```


REST 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": "ABC"
    }
  }
}
```

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": "ABC"
  }
 },
 "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"
}
```


REST 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": "ABC"
    }
  }
}
```


Response to a Successful Request


```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/697691643371622800 3955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697691643458"
  },
  "id": "697691643371622800 3955",
```

```
  "orderInformation": {
   "amountDetails": {
    "authorizedAmount": "102.21",
    "currency": "ABC"
   }
  },
  "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 83
- *Merchant-Initiated No-Show Transaction with TMS* on page 87

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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa

- Visa Electron

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

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": "ABC"
     }
  },
  "paymentInformation": {
     "card": {
        "expirationYear": "2031",
        "number": "4111xxxxxxxxxxx",
        "expirationMonth": "12"
     }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
     "authReversal": {
        "method": "POST",
        "href": "/pts/v2/payments/6534214295466223903006/reversals"
     },
     "self": {
        "method": "GET",
        "href": "/pts/v2/payments/653421429546622390306"
     },
     "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/6534214295466223903006/captures"
     }
  },
  "clientReferenceInformation": {
     "code": "1653421429522"
  },
  "id": "653421429546622390306",
  "orderInformation": {
     "amountDetails": {
        "authorizedAmount": "150.00",
        "currency": "ABC"
     }
  },
  "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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta

- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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 Required 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.**

REST 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": "ABC"
    },
    "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/697692283045693400 3954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976922830456934003954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697692283160"
  },
  "id": "6976922830456934003954",
  "orderInformation": {
```

```
    "amountDetails": {
     "authorizedAmount": "102.21",
     "currency": "ABC"
    }
   },
   "paymentAccountInformation": {
    "card": {
     "type": "001"
    }
   },
   "paymentInformation": {
    "tokenizedCard": {
     "type": "001"
    },
    "instrumentIdentifier": {
     "id": "7010000000016241111",
     "state": "ACTIVE"
    },
    "card": {
     "type": "001"
    }
   },
   "pointOfSaleInformation": {
    "terminalId": "111111"
   },
   "processingInformation": {
    "paymentSolution": "015"
   },
   "processorInformation": {
    "paymentAccountReferenceNumber": "V001001302229816966750 4231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
     "code": "X",
     "codeRaw": "I1"
    }
   },
   "reconciliationId": "62700184NNMR6XFK",
   "status": "AUTHORIZED",
   "submitTimeUtc": "2023-10-19T05:11:23Z"
}
```


REST 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": "ABC"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
     "method": "POST",
     "href": "/pts/v2/payments/6976917718796256603955/reversals"
    },
    "self": {
     "method": "GET",
     "href": "/pts/v2/payments/697691771879625660 3955"
    },
    "capture": {
     "method": "POST",
     "href": "/pts/v2/payments/6976917718796256603955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697691771976"
  },
  "id": "6976917718796256603955",
  "orderInformation": {
    "amountDetails": {
     "authorizedAmount": "102.21",
     "currency": "ABC"
    }
  },
  "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": "V001001302229816966750 4231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62700629BNN13VGW",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T05:02:52Z"
}
```

REST 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": "ABC"
    }
  }
```

```
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/697691643371622800 3955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976916433716228003955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697691643458"
  },
  "id": "697691643371622800 3955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "ABC"
    }
  },
  "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": "V001001302229816966750 4231315",
    "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 Payment with 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 Required 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": "ABC"
      }
    },
    "paymentInformation": {
      "card": {
        "expirationYear": "2031",
        "number": "4111xxxxxxxxxxx",
        "expirationMonth": "12"
      }
    }
}
```

Response to a Successful Request

```
{
    "_links": {
      "authReversal": {
        "method": "POST",
        "href": "/pts/v2/payments/6528187198946076303004/reversals"
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/652818719894607630304"
      },
      "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/6528187198946076303004/captures"
      }
    },
    "clientReferenceInformation": {
      "code": "1652818719876"
    },
    "id": "652818719894607630304",
    "orderInformation": {
      "amountDetails": {
        "authorizedAmount": "100.00",
        "currency": "ABC"
      }
    },
    "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
- Carta Si
- Cartes Bancaires
- Dankort

- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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 Required 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": "ABC"
    },
    "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": "ABC"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
```

```
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "paymentAccountReferenceNumber": "V001001302229816966750 4231315",
    "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 121.

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 202.

REST 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": "ABC"
  },
  "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/709406002003624180 3954/captures"
    }
   },
   "clientReferenceInformation": {
    "code": "1709406002076"
   },
   "id": "709406002003624180 3954",
   "orderInformation": {
    "amountDetails": {
     "authorizedAmount": "102.21",
     "currency": "ABC"
    }
   },
   "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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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 Subsequent Installment 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. 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 MIT

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": "ABC"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxx",
```

```
        "expirationMonth": "12"
      }
    }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/653082471004680930400 2"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "ABC"
    }
  },
  "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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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 MIT

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: 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": "ABC"
  },
  "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/653082471004680930402"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "ABC"
    }
  },
  "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 158 and *Mastercard Subscription Payments* on page 167.

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
- Mastercard
- Visa

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 158 and *Mastercard Subscription Payments* on page 167.

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

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: Customer-Initiated Recurring Payment Authorization 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": "ABC"
      }
    },
    "paymentInformation": {
      "card": {
        "expirationYear": "2031",
        "number": "4111xxxxxxxxxxx",
        "expirationMonth": "12"
      }
    }
}
```

Response to a Successful Request

```
{
    "_links": {
      "authReversal": {
        "method": "POST",
        "href": "/pts/v2/payments/6528187198946076303004/reversals"
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/652818719894607630304"
      },
      "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/6528187198946076303004/captures"
      }
    },
    "clientReferenceInformation": {
      "code": "1652818719876"
    },
    "id": "652818719894607630304",
    "orderInformation": {
      "amountDetails": {
        "authorizedAmount": "100.00",
        "currency": "ABC"
      }
    },
    "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
- Mastercard
- Visa

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 158 and *Mastercard Subscription Payments* on page 167.

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": "ABC"
    },
    "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/697685813410610570 3954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976858134106105703954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697685813462"
  },
  "id": "6976858134106105703954",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "ABC"
    }
  },
  "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 158 and *Mastercard Subscription Payments* on page 167.

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 Payment with TMS* on page 149.

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 202.

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": "ABC"
  },
  "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/709406002003624180 3954"
    },
    "capture": {
     "method": "POST",
     "href": "/pts/v2/payments/709406002003624180 3954/captures"
    }
   },
   "clientReferenceInformation": {
    "code": "1709406002076"
   },
   "id": "709406002003624180 3954",
   "orderInformation": {
    "amountDetails": {
     "authorizedAmount": "102.21",
     "currency": "ABC"
    }
   },
   "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": "701000000000 16241111"
    }
   }
  }
```

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
- Mastercard
- Visa

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 158 and *Mastercard Subscription Payments* on page 167.

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** `2029`**.**

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

processingInformation.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	

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 167 and *Mastercard Standing Order Payments* on page 158.

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": "ABC"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxx",
      "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": "ABC"
      }
    },
    "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 Payment 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 158 and *Mastercard Subscription Payments* on page 167.

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 `2029`.**

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 167 and *Mastercard Standing Order Payments* on page 158.

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": "4111xxxxxxxxxxx"
    }
  },
```

```
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "102.21",
      "currency": "ABC"
    },
    "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/653082471004680930402"
        },
        "capture": {
            "method": "POST",
            "href": "/pts/v2/payments/6530824710046809304002/captures"
        }
    },
    "clientReferenceInformation": {
        "code": "1653082470983"
    },
    "id": "6530824710046809304002",
    "orderInformation": {
        "amountDetails": {
            "authorizedAmount": "100.00",
            "currency": "ABC"
        }
    },
    "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": "ABC"
    }
  }
}
```


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": "ABC"
    }
   },
   "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": "ABC"
    }
  }
}
```


Response to a Successful Request


```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6974846967476340503955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/69748469674763405039 55"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6974846967476340503955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_3"
  },
  "id": "6974846967476340503955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "ABC"
```

```
    }
   },
   "paymentAccountInformation": {
    "card": {
     "type": "001"
    }
   },
   "paymentInformation": {
    "tokenizedCard": {
     "type": "001"
    },
    "card": {
     "type": "001"
    }
   },
   "pointOfSaleInformation": {
    "terminalId": "111111"
   },
   "processingInformation": {
    "paymentSolution": "015"
   },
   "processorInformation": {
    "paymentAccountReferenceNumber": "V001001302229816966750 4231315",
    "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": "ABC"
      }
    },
    "paymentInformation": {
      "card": {
        "expirationYear": "2031",
        "number": "5555xxxxxxxxxxx",
        "expirationMonth": "12"
      }
    }
}
```

Response to a Successful Request

```
{
    "_links": {
      "authReversal": {
        "method": "POST",
        "href": "/pts/v2/payments/6530824710046809304002/reversals"
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/653082471004680930400 2"
      },
      "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/6530824710046809304002/captures"
      }
    },
    "clientReferenceInformation": {
      "code": "1653082470983"
    },
    "id": "6530824710046809304002",
    "orderInformation": {
      "amountDetails": {
        "authorizedAmount": "100.00",
        "currency": "ABC"
      }
```

```
  },
  "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": "ABC"
  },
  "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/706495941148670650 3954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/70649594114867065 03954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/706495941148670650 3954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1706495941197"
 },
 "id": "7064959411486706503954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "100.00",
   "currency": "ABC"
  }
 },
 "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": "ABC"
      }
    },
    "paymentInformation": {
      "card": {
        "expirationYear": "2031",
        "number": "4111xxxxxxxxxxx",
        "expirationMonth": "12"
      }
    }
}
```

Response to a Successful Request

```
{
    "_links": {
      "authReversal": {
        "method": "POST",
        "href": "/pts/v2/payments/6530824710046809304002/reversals"
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/653082471004680930400 2"
      },
      "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/6530824710046809304002/captures"
      }
    },
    "clientReferenceInformation": {
      "code": "1653082470983"
    },
    "id": "6530824710046809304002",
    "orderInformation": {
      "amountDetails": {
        "authorizedAmount": "100.00",
        "currency": "ABC"
      }
    },
    "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": "ABC"
  },
  "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/706494684625641010 3954"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/706494684625641010 3954/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1706494684667"
  },
  "id": "706494684625641010 3954",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "ABC"
    }
  },
  "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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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

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": "ABC"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/652818719894607630304"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1652818719876"
  },
  "id": "652818719894607630304",
```

```
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "ABC"
    }
  },
  "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
- Carta Si
- Cartes Bancaires

- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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": "ABC"
  },
  "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": "ABC"
   }
  },
  "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 191.

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": "ABC"
  },
  "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/709406002003624180 3954/reversals"
  },
  "self": {
   "method": "GET",
   "href": "/pts/v2/payments/7094060020036241803954"
  },
  "capture": {
   "method": "POST",
   "href": "/pts/v2/payments/709406002003624180 3954/captures"
  }
 },
 "clientReferenceInformation": {
  "code": "1709406002076"
 },
 "id": "7094060020036241803954",
 "orderInformation": {
  "amountDetails": {
   "authorizedAmount": "102.21",
   "currency": "ABC"
  }
 },
 "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 Payment 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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

Endpoint

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

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

Required Fields for a Subsequent MIT 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": "ABC"
      }
    },
    "paymentInformation": {
      "card": {
        "expirationYear": "2031",
        "number": "4111xxxxxxxxxxx",
        "expirationMonth": "12"
      }
    }
}
```

Response to a Successful Request

```
{
    "_links": {
      "authReversal": {
        "method": "POST",
        "href": "/pts/v2/payments/6530824710046809304002/reversals"
      },
      "self": {
        "method": "GET",
        "href": "/pts/v2/payments/653082471004680930402"
      },
      "capture": {
        "method": "POST",
        "href": "/pts/v2/payments/6530824710046809304002/captures"
      }
    },
    "clientReferenceInformation": {
      "code": "1653082470983"
    },
    "id": "6530824710046809304002",
    "orderInformation": {
```

```
      "amountDetails": {
        "authorizedAmount": "100.00",
        "currency": "ABC"
      }
    },
    "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
- Carta Si
- Cartes Bancaires
- Dankort
- Delta
- Eurocard
- JCB
- Maestro (UK Domestic)
- Mastercard
- Visa
- Visa Electron

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

REST 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": "ABC"
  },
  "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": "ABC"
    }
   },
   "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"
}
```

REST Example: MIT Unscheduled COF Payment with TMS Payment Instrument

Request

```
{
 "processingInformation": {
  "commerceIndicator": "internet"
 },
 "paymentInformation": {
  "paymentInstrument": {
   "id": "080AE120369A7947E063A2598D0A718F"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "ABC"
  }
 }
}
```

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": "ABC"
  }
 },
 "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": "V001001302229816966750 4231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
     "code": "X",
     "codeRaw": "I1"
    }
   },
   "reconciliationId": "62699372XNMR85HS",
   "status": "AUTHORIZED",
   "submitTimeUtc": "2023-10-19T04:18:50Z"
}
```


REST Example: MIT Unscheduled COF Payment with TMS Customer

Request


```
{
 "processingInformation": {
  "commerceIndicator": "internet"
 },
 "paymentInformation": {
  "customer": {
   "id": "080AC9AB60C92AA2E063A2598D0A0C74"
  }
 },
 "orderInformation": {
  "amountDetails": {
   "totalAmount": "102.21",
   "currency": "ABC"
  }
 }
```

```
}
```

Response to a Successful Request

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6976889582016147703955/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/697688958201614770 3955"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6976889582016147703955/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1697688958296"
  },
  "id": "6976889582016147703955",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "102.21",
      "currency": "ABC"
    }
  },
  "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": "V001001302229816966750 4231315",
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "62699842BNN13VA0",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-10-19T04:15:58Z"
}
```

Reference Information

This section contains this helpful reference information when processing credentialed transactions.

Payer Authentication Values

This section describes the possible payer authentication values you can include in the **processingInformation.commerceIndicator** request field.
The level of security in payer authentication is indicated by the two-digit e-commerce indicator (ECI) that is assigned to the transaction. These values have text equivalents that are assigned to the **processingInformation.commerceIndicator** field.

The American Express, China UnionPay, Diners, Discover, 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	China UnionPay	American Express
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

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.**