

Card Present Connect | Mass Transit

REST API
Visa Platform Connect



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

© 2020. Cybersource Corporation. All rights reserved. Cybersource Corporation (“Cybersource”) furnishes this document and the software described in this document under the applicable agreement between the reader of this document (“You”) and Cybersource (“Agreement”). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by Cybersource. Cybersource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of Cybersource.

Restricted Rights Legends

For Government or defense agencies: Use, duplication, or disclosure by the Government or defense agencies is subject to restrictions as set forth in the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

For civilian agencies: Use, reproduction, or disclosure is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in Cybersource Corporation's standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

Trademarks

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

Confidentiality Notice

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

By accepting this document, you acknowledge that the information contained herein (the “Information”) is confidential and subject to the confidentiality restrictions contained in Visa's operating regulations and/or other confidentiality agreements, which limit 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: 25.02

Contents

Card Present Connect Mass Transit Developer Guide	7
Recent Revisions to This Document.....	8
VISA Platform Connect: Specifications and Conditions for Resellers/Partners.....	9
Introduction to Card Present Connect Mass Transit	10
Rider Benefits.....	10
Transit System Benefits.....	11
Mass Transit Terminology.....	11
Mass Transit Prerequisites.....	13
Mass Transit Validation and Certification.....	14
Message-Level Validation.....	14
EMV L3 Certification.....	14
Mass Transit Transactions.....	15
Aggregated.....	15
Debt Recovery.....	17
Card Types and Transaction Models.....	17
Aggregated Model with the Token Management Service.....	18
Multiple Accounts for Processing Functions.....	20
Journey History Service.....	21
American Express Mass Transit Model.....	22
Discover Mass Transit Model.....	25
Mastercard Mass Transit Model.....	26
Visa Mass Transit Model.....	28
Mass Transit Transaction Workflows	31
American Express Pay As You Go Workflow.....	31
Discover Pay As You Go Workflow.....	33
Mastercard Pay As You Go Workflow.....	35
Visa Mobility and Transport Transaction Workflow.....	36
Additional Workflows.....	38
Near Real-Time Workflow.....	38
Fare Calculation and Submission Workflow.....	40
Debt Recovery Workflows.....	42
Mass Transit Payment Services Using EMV and Card Data	46

Transaction Types.....	48
Required Fields for Authorizations with EMV Data.....	52
American Express Account Status Check Authorization with EMV Data.....	54
Required Fields for a American Express Account Status Check AVR Authorization with EMV Data Using the REST API.....	54
REST Example: American Express Account Status Check Authorization with EMV Data.....	55
American Express Delayed Online Authorization with EMV Data.....	58
Required Fields for a American Express Delayed Online Authorization with EMV Data Using the REST API.....	58
REST Example: American Express Delayed Online Authorization with EMV Data.....	59
Discover Authorization with EMV Data.....	61
Required Fields for a Discover Authorization with EMV Data Using the REST API.....	61
REST Example: Discover Authorization with EMV Data.....	62
Mastercard Authorization with EMV Data.....	64
Required Fields for a Mastercard Authorization with EMV Data Using the REST API.....	64
REST Example: Mastercard Authorization with EMV Data.....	65
Visa Account Verification Request (AVR) with EMV Data.....	67
Required Fields for a Visa AVR Authorization with EMV Data Using the REST API.....	67
REST Example: Visa AVR Authorization with EMV Data.....	68
Required Fields for Sales with EMV Data.....	71
Discover Sale with EMV Data.....	72
Required Fields for a Discover Sale with EMV Data Using the REST API.....	72
REST Example: Discover Sale with EMV Data.....	73
Visa Deferred Sale with EMV Data.....	76
Required Fields for a Visa Deferred Sale with EMV Data Using the REST API.....	76
REST Example: Visa Deferred Sale with EMV Data.....	77
Required Fields for an Authorization for Debt Recovery.....	80
Tap-Initiated Authorization for Debt Recovery with EMV Data.....	82
Required Fields for a Tap-Initiated Authorization for Debt Recovery with EMV Data Using the REST API.....	82
REST Example: Tap-Initiated Authorization for Debt Recovery with EMV Data.....	83
Merchant-Initiated Authorizations for Debt Recovery with Stored Card Data.....	85
Required Fields for a Merchant-Initiated Authorization for Debt Recovery with Stored Card Data Using the REST API.....	85
REST Example: Merchant-Initiated Authorization for Debt Recovery with Stored Card Data.....	86
Required Fields for a Sale for Debt Recovery.....	87
Tap-Initiated Sale for Mastercard Debt Recovery with EMV Data.....	90
Required Fields for a Tap-Initiated Sale for Mastercard Debt Recovery with EMV Data Using the REST API.....	90

REST Example: Tap-Initiated Sale for Mastercard Debt Recovery with EMV Data.....	91
Tap-Initiated Sale for Visa Debt Recovery with EMV Data.....	93
Required Fields for a Tap-Initiated Sale for Visa Debt Recovery with EMV Data Using the REST API.....	93
REST Example: Tap-Initiated Sale for Visa Debt Recovery with EMV Data.....	94
Merchant-Initiated Sale for Debt Recovery with Stored Card Data.....	96
Required Fields for a Merchant-Initiated Sale for Debt Recovery with Stored Card Data Using the REST API.....	96
REST Example: Merchant-Initiated Sale for Debt Recovery with Stored Card Data.....	97
Mass Transit Payment Services Using TMS Tokens.....	100
Mass Transit Mastercard Authorization with a Token.....	101
Required Fields for a Mastercard Authorization with a Token Using the REST API.....	101
REST Example: Performing a Mastercard Authorization with a Token.....	102
Mass Transit Visa Account Verification Request (AVR) with a Token.....	104
Required Fields for a Visa AVR Authorization with a Transient Token Using the REST API.....	105
REST Example: Performing a Visa AVR Authorization with a Transient Token.....	105
Mass Transit Visa Deferred Sale with a Token.....	107
Required Fields for a Visa Deferred Sale with a Token Using the REST API.....	107
REST Example: Performing a Visa Deferred Sale with a Token.....	108
Mass Transit Authorizations for a Tap-Initiated Debt Recovery with a Token.....	110
Required Fields for a Tap-Initiated Authorization for Debt Recovery with a Token Using the REST API.....	110
REST Example: Performing a Tap-Initiated Authorization for Debt Recovery with a Token.....	111
Mass Transit Merchant-Initiated Authorizations for Debt Recovery with a Token.....	113
Required Fields for a Merchant-Initiated Authorization for Debt Recovery with a Token Using the REST API.....	113
REST Example: Performing a Merchant-Initiated Authorization for Debt Recovery with a Token.....	114
Mass Transit Tap-Initiated Sale for Debt Recovery with a Token.....	116
Required Fields for a Tap-Initiated Sale for Debt Recovery with a Token Using the REST API.....	116
REST Example: Performing a Tap-Initiated Sale for Debt Recovery with a Token.....	117
Mass Transit Merchant-Initiated Sale for Debt Recovery with a Token.....	119
Required Fields for a Merchant-Initiated Sale for Debt Recovery with a Token Using the REST API.....	119
REST Example: Performing a Merchant-Initiated Sale for Debt Recovery with a Token.....	120
Mass Transit Stand-Alone Credit with a Token.....	122
Required Fields for a Stand-Alone Credit with a Token Using the REST API.....	123
REST Example: Performing a Stand-Alone Credit with a Token.....	123

Mass Transit Follow-On Payment Services	125
Required Fields for Mass Transit Captures.....	125
Capture an Authorization.....	126
Required Fields for a Mass Transit Capture.....	126
REST Example: Capture an Authorization.....	126
Required Fields for Mass Transit Reversals.....	127
Authorization Reversal.....	128
Required Fields for a Mass Transit Authorization Reversal.....	128
REST Example: Reversing a Mass Transit Authorization.....	128
Timeout Reversal.....	129
Required Fields for a Timeout Reversal.....	129
REST Example: Timeout Reversal.....	130
Required Field for Timeout Voids.....	131
Timeout Void.....	131
Required Field for a Timeout Void.....	132
REST Example: Timeout Void.....	132
Mass Transit Token Management Services	134
Creating a Token.....	134
Required Fields for Creating a Token Using the REST API.....	136
REST Example: Create a Token.....	136
Retrieving Transient Token Details.....	137
REST Example: Retrieve Transient Token Details Using the REST API.....	137
Retrieving an Instrument Identifier Details.....	138
REST Example: Retrieving Instrument Identifier Details Using the REST API.....	138
Retrieving Card Hash Details.....	139
REST Example: Retrieving Card Hash Details Using the REST API.....	139
Retrieving Payment Instrument Details.....	140
REST Example: Retrieving Payment Instrument Details Using the REST API.....	141
Deleting an Instrument Identifier.....	142
REST Example: Deleting an Instrument Identifier Using the REST API.....	142
Deleting a Card Hash.....	143
REST Example: Deleting a Card Hash.....	143
Transit Test Cases	144

Card Present Connect | Mass Transit Developer Guide

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

Audience and Purpose

This guide is written for application developers who want to integrate payment processing for mass transit fare collection systems. These services are available using only the REST API.

Implementing these services requires software development skills and knowledge and understanding of the card scheme mass transit rules. You must write code that uses the REST API request and response fields to integrate the payment services into your existing mass transit fare collection system.

Conventions

These special statements are used in this document:

Important

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

Warning

A Warning contains information or instructions, which, if not

heeded, can result in a security risk, irreversible loss of data, or significant cost in time or revenue or both.

Related Documentation

Visit the [Cybersource documentation hub](#) to find additional technical documentation. Contact the card schemes for these technical documents:

- Mastercard Global Transit Implementation Guide
- Mastercard Transit Solutions
- Mastercard Transit Terminal Requirements
- Visa Contactless Transit Implementation Guide
- Visa Contactless Transit Terminal Implementation Guide
- Visa Transforming Urban Mobility

Customer Support

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

<http://support.visaacceptance.com>

Recent Revisions to This Document

25.02

Transaction Types

Updated the first ride risk (FRR) transaction type descriptions for field values `TransitDA FRR MIT DR auth` and `TransitDA FRR tap DR auth`. See [Transaction Types](#) on page 48.

25.01

Discover

Added support for Discover cards in the U.S. See these sections:

- [Discover Aggregated Transactions](#) on page 16
- [Discover Mass Transit Model](#) on page 25
- [Discover Pay As You Go Workflow](#) on page 33

- [Discover Authorization with EMV Data](#) on page 61
- [Discover Sale with EMV Data](#) on page 72

24.07

Updated the journey history information. See [Journey History Service](#) on page 21.
Added lists of required fields for services using EMV and card data. See [Mass Transit Payment Services Using EMV and Card Data](#) on page 46.

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 Card Present Connect | Mass Transit

Card Present Connect | Mass Transit is the Cybersource solution for processing transactions using card scheme mass transit models. Cybersource mass transit processing is based on the global standards established by the card schemes for a reliable, scalable, and secure EMV contactless fare collection system.

Supported card types:

- American Express
- Discover (U.S. only)
- Mastercard
- Visa

Rider Benefits

Mass transit riders should experience these benefits:

- Retail-like experience where the contactless device and the reader communicate over the contactless interface to perform a contactless transaction.
- Fast, contactless tap to enter and exit.
- Payment card data protection.
- Single, combined payment for multiple trips during a set period, usually at the end of the day.
- Ability to request journey history and corresponding receipt.
- Travel fare total on payment card statements.
- Similar fare collection experience on multiple transit systems in other cities and countries.

Transit System Benefits

Mass transit systems will benefit in these ways:

- Lower ticketing overhead without ticket booths or paper tickets.
- Ability to track riders when they tap to enter and exit.
- Flexible fare management
 - Riders pay as they go.
 - Fares are aggregated for one payment transaction each travel period.
- Merchant protection.
 - Encrypted payment data.
 - First Ride Risk in supported regions.
 - Debt Recovery.

Mass Transit Terminology

Account Verification Request (AVR)	Zero amount authorization request that you send to determine whether the payment card is valid.
Aggregated	Transaction in which you calculate the fare based on multiple contactless card taps for trips during a predetermined time period, usually 24 hours, processed as a single transaction.
Back office	A component within your transit systems that processes the taps received from transit readers, and that performs any or all journey construction, fare calculation, risk management, and payment processing.
Card hash	One-way hash token ID of the payment card data that is used to maintain the deny list.
Combined data authentication (CDA)	Authentication technique that uses a combination of card and transaction data.
Deferred authorization	Combined authorization and capture request, also known as a sale, for aggregated fare payments.
Deny list	List of cards that failed ODA because of an unsuccessful AVR or transit payment. It is used for blocking cards that have not been accepted for travel within your

	transit system when you are processing aggregated payments, such as the Mastercard PAYG and Visa MTT models.
Deny list manager	Manages the deny list and distributes it to the validators.
First ride risk debt recovery	Under specific and limited conditions established by the card schemes, you can recover the cost of the first ride by capturing a declined authorization. For details, refer to each of the card scheme's rules for mass transit chargeback thresholds and protection.
Instrument identifier token	Token Management Service token that stores the payment card number.
Journey construction	The process of analyzing individual taps received from transit readers and forming logical journeys performed by cardholders.
Mobility and Transport Transaction (MTT)	Visa model for contactless mass transit payments for single or multiple modes of transportation, which includes fixed, distance-based, and time-based fares.
Offline data authentication (ODA)	EMV security feature in which payment cards are authenticated offline. ODA is necessary so that cardholders can quickly tap and enter the transit system. It is used for aggregated transactions, such as Mastercard PAYG and Visa MTT.
Pay As You Go (PAYG) for Mastercard	Mastercard model in which the fare is not known when the cardholder taps their card for travel. All cardholder taps are recorded to calculate the fare and process an aggregated payment.
Payment instrument token	Token Management Service token that stores the instrument identifier token, card expiration date, and billing address.
Retail	Transaction in which you process the payment as a standard contactless retail payment.
Tap	Refers to the act of presenting a contactless card at a validator.
Ticket inspection	Process in which ticket inspectors verify compliance with fare policies by checking

	paper tickets or using a portable terminal to read the payment card.
Ticket inspectors	Transit employees who travel on the transit system to verify passenger travel status.
Transient token	Unique, time-limited token ID that is associated with the tokens created by TMS. The validator forwards this ID to your back office to use for payment transactions and to manage tokens. Cybersource automatically deletes this token after seven days.
Travel period	Period of time during which a traveler can make multiple taps in and out of the transit system, before you submit the final payment transaction for the aggregated amount.
Validator	EMV contactless card-present terminal located at an automated turnstile device where cardholders tap their card to enter, and optionally exit from, the transit system. Before allowing the cardholder to enter the transit system, the validator checks the deny list to ensure that the card has not failed ODA.

Mass Transit Prerequisites

Before integrating Cybersource services for mass transit, you must have these systems in place:

- Merchant account with an acquirer that is enabled for mass transit transactions on Visa Platform Connect.
- Cybersource account for payment services.
- Payment technology provider (PTP) that is integrated with Cybersource and can perform message-level validation (MLV).
- EMV Level 1 certified transit terminals and EMV Level 2 certified software in preparation for EMV Level 3 Certification.

Related concepts

- [Message-Level Validation on page 14](#)
- [EMV L3 Certification on page 14](#)

Mass Transit Validation and Certification

Work with your payment technology provider (PTP) to allocate time to complete the message-level validation (MLV) and EMV Level 3 certification with your mass transit fare processing system. You must pass MLV before beginning EMV Level 3 certification. You must complete validation and certification before your system can go live.

Message-Level Validation

Message-level validation (MLV) is a script-based field-level validation against Cybersource specifications.

Your PTP uses amount-based test triggers to send transactions into a test environment and the Visa Certification Management System for decryption. The test results are XML or RESTful output, Business Center test transactions, and log prints.

Cybersource validates the results with these activities:

- Cross edit checks
- Data element validation
- Interchange compliance
- Data mapping validation

Related reference

- Transit Test Cases on page 144

EMV L3 Certification

This topic is an overview of the Level 3 accreditation with Cybersource and Visa Platform Connect.

For details on how to design an EMV Level 3 certified payment application, find EMV Book 3 Application Specification at:

<https://www.emvco.com/specifications/>

Certification is a formal process to validate the device and application compliance with card scheme acceptance regulations. The certification team uses a brand test tool and simulator. The process involves:

- Using a card simulator such as UL, ICC, or Fime.
- Failed case analysis and resolution.
- For Mastercard certification, your PTP submits results to Mastercard and pays the costs for approved partners that Mastercard uses.
- For Visa certification, Cybersource submits results to Visa.
- Waivers from the card schemes for exceptions.
- Card schemes responses or Letter of Approval (LOA) to signify acceptance and Level 3 certification.

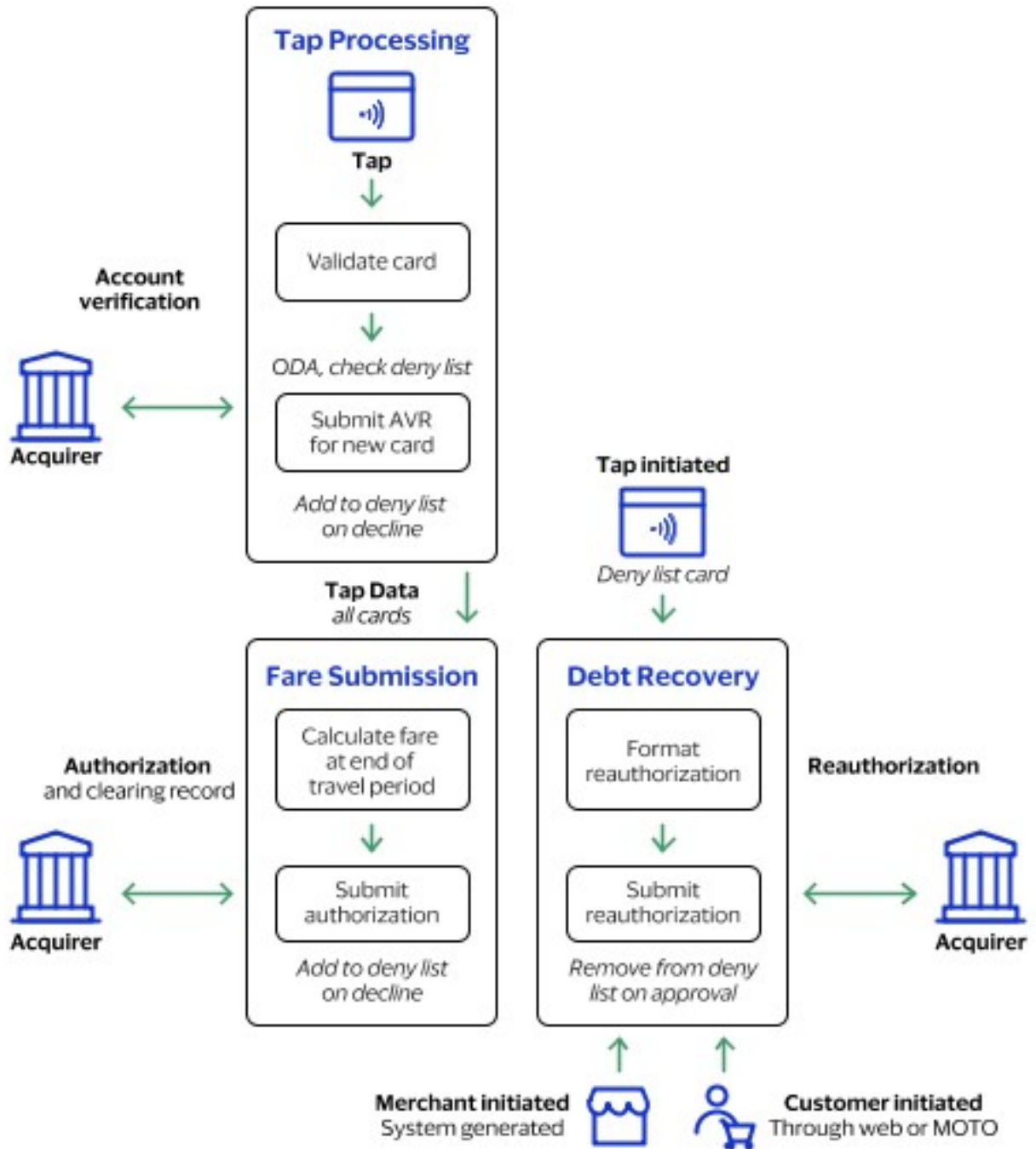
The processes and support for Global Card Present Connect projects and direct merchant and acquirer projects are different, but the timelines are essentially the same.

Mass Transit Transactions

Cybersource offers two types of transactions for mass transit fare collection and management, aggregated and debt recovery.

Aggregated

This type of transaction uses a contactless terminal at points of access to the transit service. The final fare charged is not always known at the time of travel. It is calculated at the end of a travel period (typically 24 hours) based on journeys made during that period.



Aggregated Transaction Flow

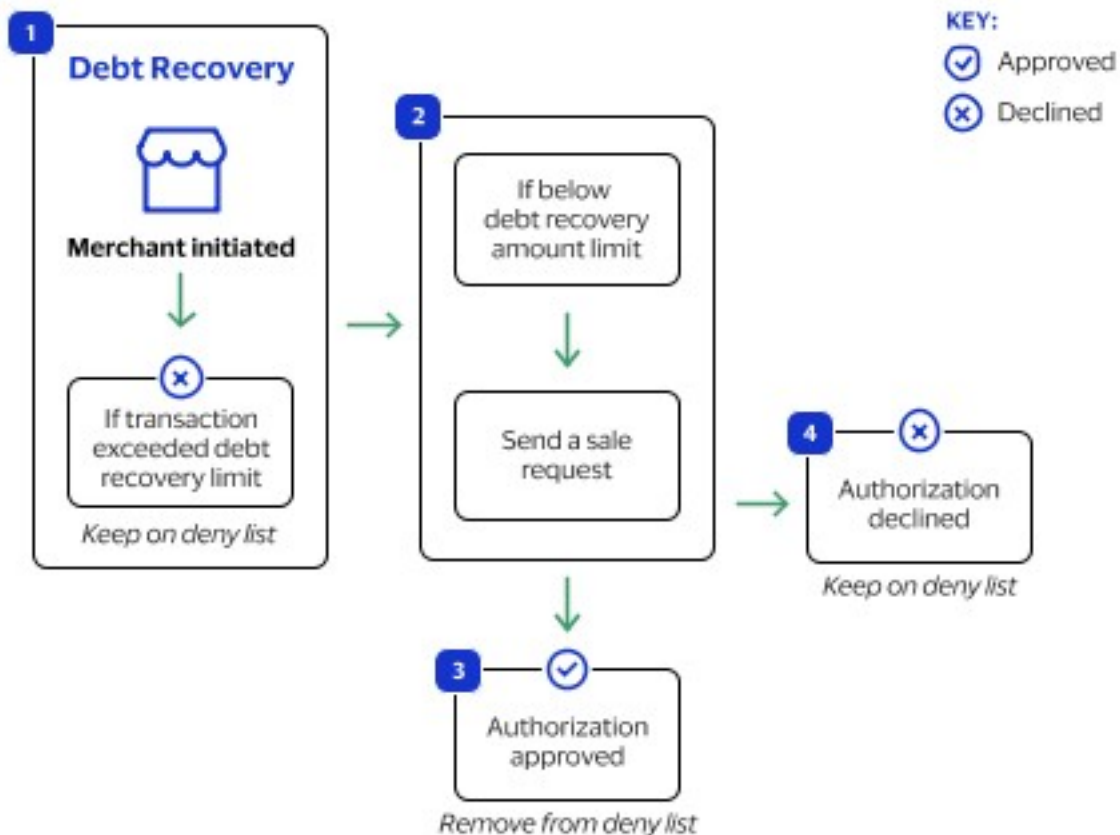
Discover Aggregated Transactions

When the cardholder taps their card for travel, initiate an aggregated transaction by requesting an authorization for 1.00 USD that includes the **processingInformation.authorizationOptions.aggregatedAuthIndicator** field set to **true**. If the issuer approves the authorization, subsequent taps do not require an authorization. The capture amount must not exceed 15.00 USD in the US, and the capture date must not exceed the aggregation duration of up to 14 days from the first tap transaction.

If the aggregate total nears the maximum aggregation amount, and the next tap will cause the aggregation amount to exceed 15.00 USD, capture the existing aggregate total. Then, authorize the current tap for 1.00 USD to begin a new aggregation cycle, and include the **processingInformation.authorizationOptions.aggregatedAuthIndicator** field set to `true`. For more details about Discover aggregated transactions, see the Discover Global Network Contactless D-PAS: Open Loop Transit Implementation Guide.

Debt Recovery

This type of transaction retrieves outstanding debt in the event of a decline of the aggregated, end-of-day transaction. A debt recovery transaction is also required in order to remove a card from a deny list. Card schemes require you to process merchant-initiated debt recovery. You can also offer tap-initiated debt recovery at the transit entry point or cardholder-initiated debt recovery online.



Merchant-Initiated Debt Recovery Flow

Card Types and Transaction Models

These card types, transaction types, and acceptance models are supported.

American Express

- Aggregated
 - Pay As You Go (PAYG) delayed authorization with the Expresspay transit policy

Discover

- Aggregated, and only supported in the U.S.
 - Pay As You Go (PAYG)

Mastercard

- Aggregated
 - Pay As You Go (PAYG)

Visa

- Aggregated
 - Mobility and Transport Transaction (MTT)

All Card Types

- Debt Recovery
 - First ride risk
 - Merchant Initiated
 - Tap Initiated
 - Cardholder Initiated

Aggregated Model with the Token Management Service

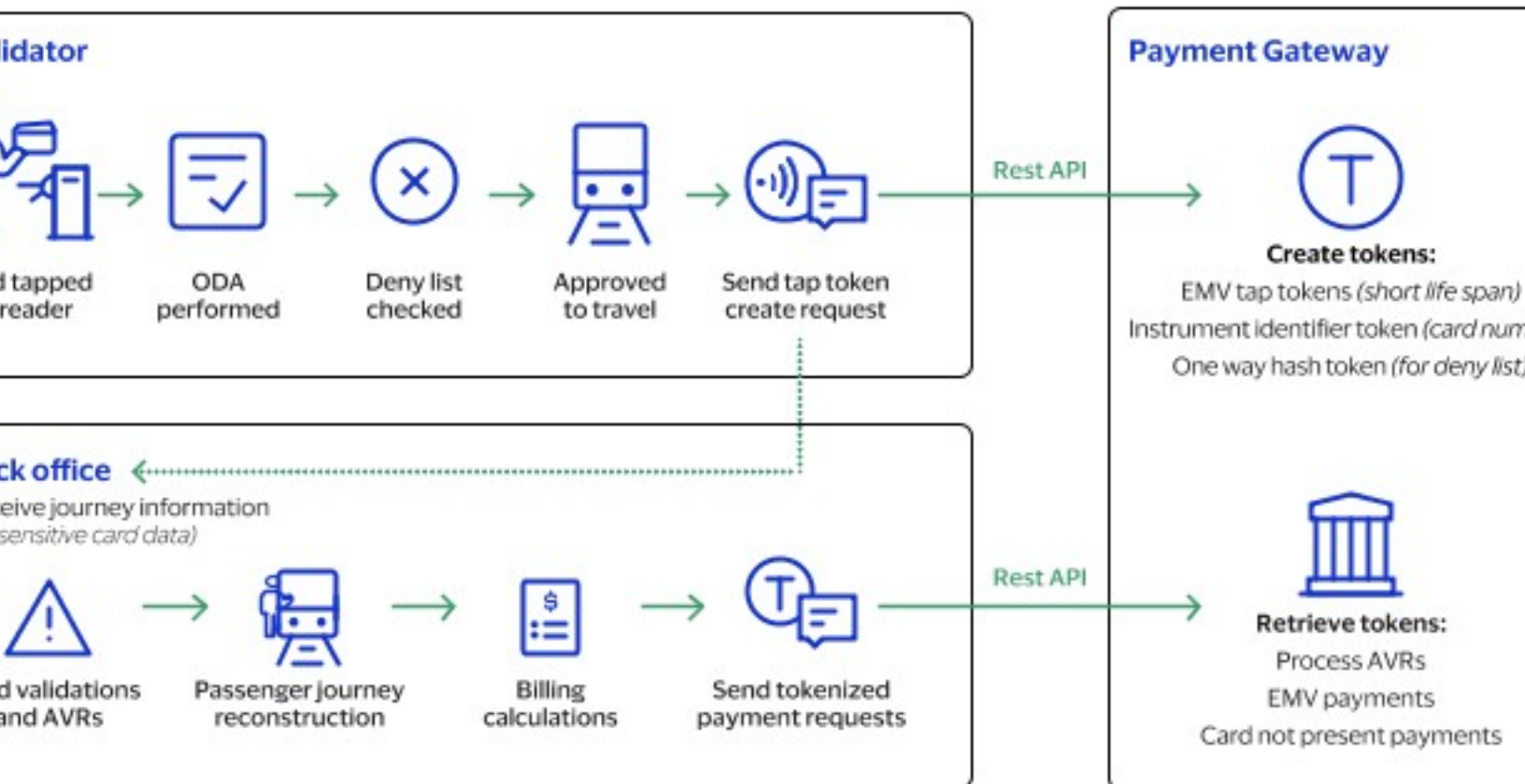
When you integrate aggregated acceptance models with the Token Management Service (TMS), customer and payment data is tokenized, securely stored, and managed by TMS, which simplifies your Payment Card Industry Data Security Standard (PCI DSS) compliance for storing, processing, or transmitting cardholder information. PCI DSS compliance requirements are for establishing and maintaining a reliable and secure payment processing environment.

You can use TMS to store these types of data:

- EMV track 2 equivalent
- EMV tag-length-value (TLV) string of tags for use during payment authorization
- Card number (TMS instrument identifier)
- Card hash value (used within deny list management systems)

TMS uses a cryptographic base derivation key (BDK) to create tokens that represent the customer and payment data. You store these tokens in your environment and databases instead of customer payment details.

The aggregated model with TMS begins with the first tap of the card.



Transit Transactions with TMS

First Tap Process with TMS

This process begins when the rider taps their card at the validator.

1. Rider taps card at the validator.
2. The terminal uses the Level 3 payment application to generate a card hash and checks the deny list for the card hash.
3. When the card hash is on the deny list, the card is not approved for travel and the terminal does not open the gate.
4. When the card hash is not on the deny list, the card is approved for travel and the terminal opens the gate and begins the account verification request (AVR) process.
5. The rider performs additional taps as required by the transit system during the travel period.

AVR Process with TMS

This process begins when the validator sends the transient token data to the back office.

1. The validator sends this tap data to TMS to tokenize the data:
 - Transient token, in the ID field
 - Card hash

- Fluid data descriptor, encoding, and value
2. TMS creates tokens of the tap data and stores the card hash with the tokens.
 3. The back office performs a verification request as required by the card scheme transit model.
 4. The back office uses the transient token ID for these requests:
 - Retrieving the token IDs for debt recovery and BIN lookup requests
 - Performing an AVR, deferred authorization, and follow-on transactions

End of Travel Period Process with TMS

This process begins at the end of the travel period.

1. At the end of the travel period, the back office calculates the fare and sends a deferred authorization, which can be a combined authorization and capture, using the transient token ID in place of the card data.
2. If the authorization fails, the back office retrieves the card hash from TMS, adds the card hash to the deny list, and begins the debt recovery process.

Debt Recovery Process with TMS

This process begins when your back office requests debt recovery.

1. In accordance with the relevant card scheme rule set, the back office requests a merchant-initiated debt recovery using the instrument identifier token ID in place of a card number.
2. When debt recovery is successful, the back office uses the card hash token to retrieve the full card hash value.
3. The back office removes the card hashes from the deny list.
4. When the debt recovery fails, the associated card hashes stay on the deny list.

Additional Documentation

[Token Management Service Developer Guide](#)

Multiple Accounts for Processing Functions

Mass transit systems can have multiple accounts for each processing function.

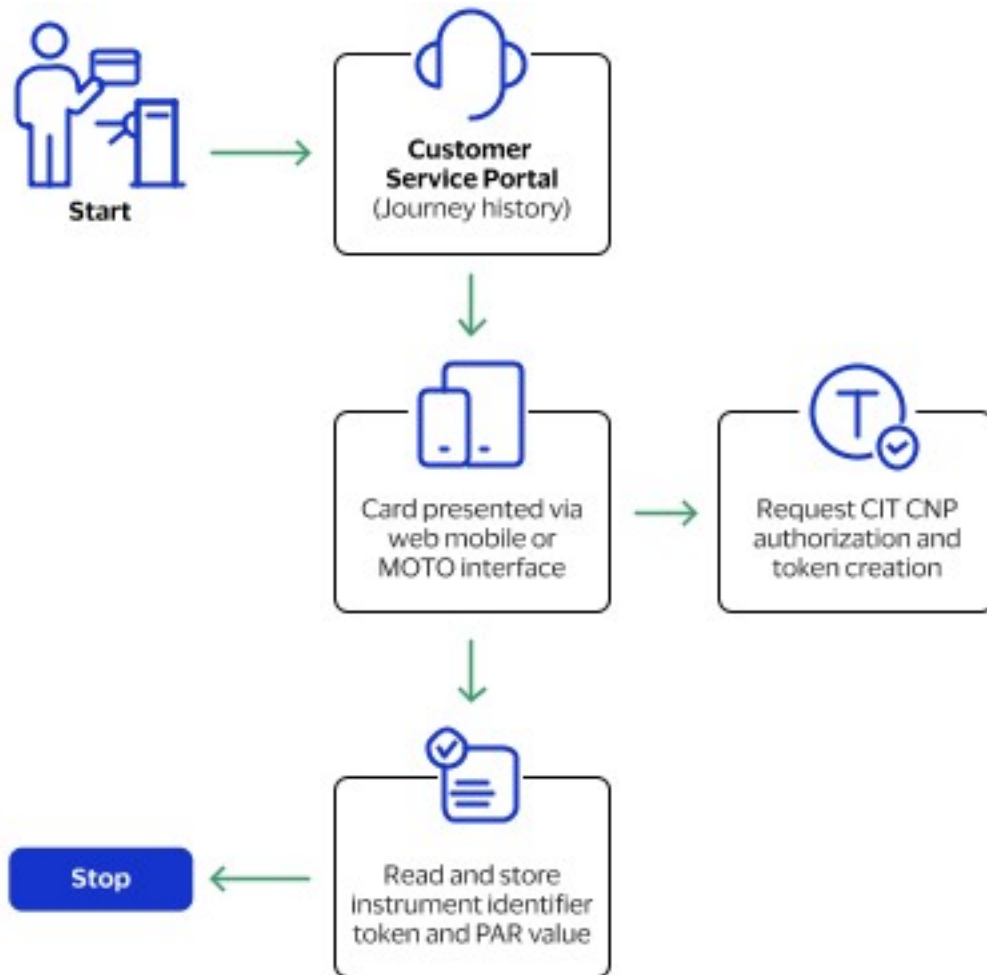
Account 1: This account processes tap token create requests only. For this option, the validators communicate directly with Cybersource. Using a separate account enables you to deploy a separate security key to the validator system.

Account 2: This account processes payment requests using tokens. You can also choose to further separate debt recovery transactions from standard payment transactions. In that case, account 2a is dedicated to standard payment transactions, and account 2b is dedicated to debt recovery transactions.

Account 3: This account processes card hash retrieval requests only. The responses include the full, unmasked card hash value. Using a separate account enables you to deploy a separate security key to the validator system.

Account 4: This account operates a customer service web portal where riders can make journey history inquiries. The riders provide the card number to a hosted payment service

(such as Cybersource Secure Acceptance) where they register as a user. Registration produces the TMS instrument identifier token and the card scheme PAR value. These values can be used by your back-office system to look up journey and billing information.



Journey History Request with Token Creation

Journey History Service

Card schemes might require transit operators to provide cardholders with a journey history service that enables the cardholder to view their journey history and receipt information. Refer to card scheme documentation for more information about each card scheme's requirements for journey history.

Payment Account Reference

Cybersource supports the use of the payment account reference (PAR) by providing the PAR value in authorization responses when a PAR is available from the card issuer. The PAR response field is **processorInformation.paymentAccountReferenceNumber**. Using the PAR enables you to track card accounts when digital devices, such as smart phones and smart watches, have a network token or DPAN that the card issuer provided to the device. You can use the PAR to provide journey history to cardholders and to match the card account FPAN and DPAN values.

Merchant Descriptor

You can use the merchant descriptor feature to produce a transaction-specific reference that cardholders can see on their card account statement. To use the merchant descriptor feature, include the **merchantInformation.merchantDescriptor.name** field in your authorization, credit, and capture requests. The value for the field must consist solely of English characters.

Before implementing transit journey history functionality, confirm with your acquirer and card schemes that the PAR and merchant descriptor features are supported.

For documentation regarding Cybersource card-not-present services that you can use to support a transit system journey history service, see the [Payments Developer Guide](#).

American Express Mass Transit Model

The American Express mass transit transaction model that is American Express PAYG with delayed authorization using Expresspay transit policy. See [American Express Pay As You Go](#) on page 22.

Characteristics of the American Express PAYG Model

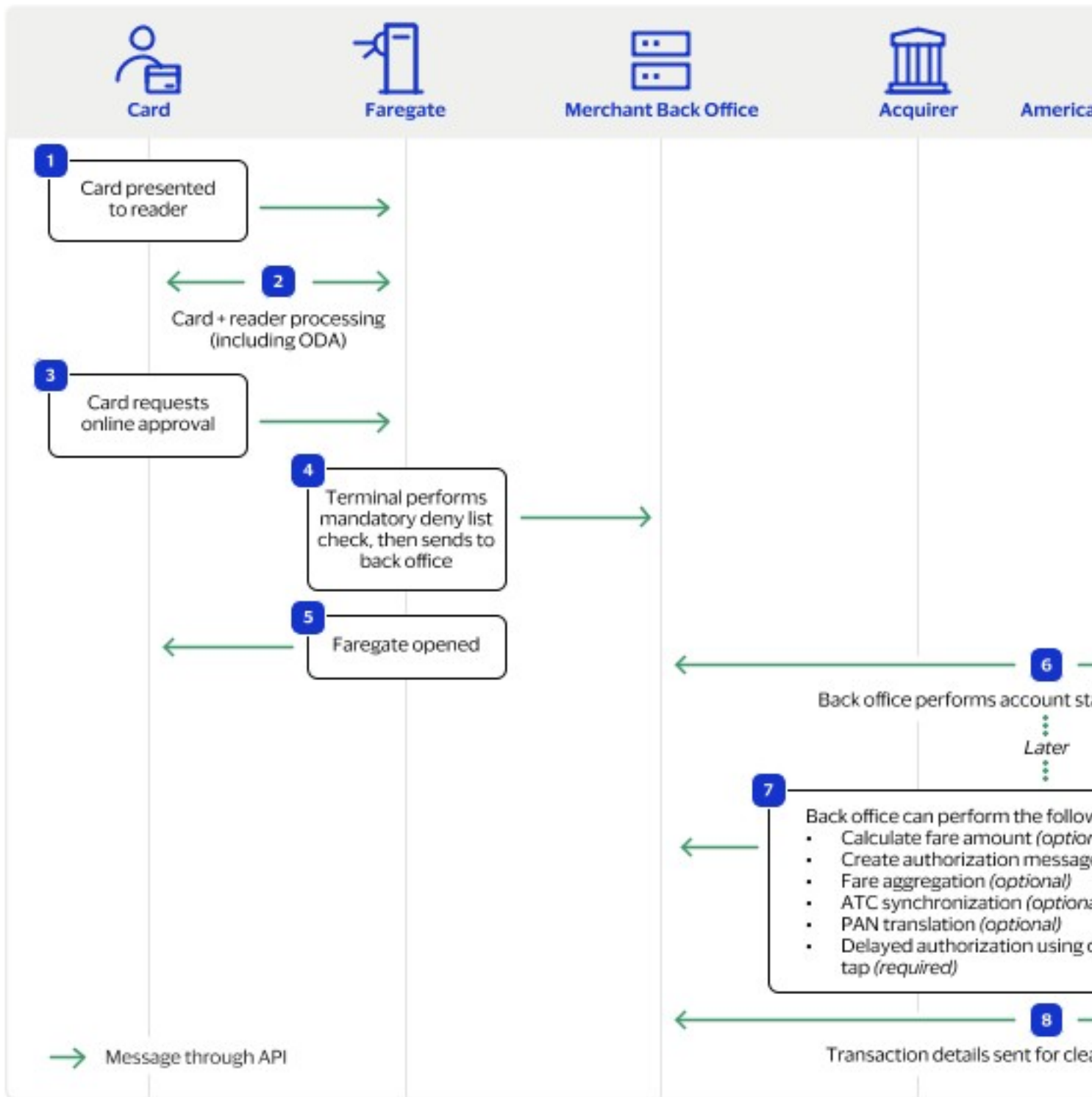
Characteristic	PAYG with Delayed Authorizations
Dedicated transit merchant category code (MCC) values	M
Population of transit access terminal indicator	M
Decline expired cards	M
Decline expired cards	M
Deny list capability	M
Transaction aggregation	O
Account status check	M
Enhanced risk mitigation	O
Application transaction counter (ATC) synchronization	O
PAN translation	O

American Express Pay As You Go

The American Express Pay As You Go (PAYG) model is a delayed authorization using the Expresspay transit policy. These are the key characteristics of this model:

- Journeys are multimodal.
- Fares are based on distance.
- Points of entry are contactless.
- Accounts are verified with an authorization for a nominal amount or the maximum fare amount.

- The deny list is checked so that cardholders with previously declined transactions can be blocked.
- Data can be authenticated offline to confirm that the card is valid.
- Multiple card taps and fares can be aggregated into a single payment.
- The deny list is automatically updated every 20 minutes.
- The back office records all trips and fares in order to reconstruct journeys and calculate the final fare.
- When the nominal amount authorization is declined, debt recovery can be performed.
- Standard follow-on payment services can be used to capture, reverse, or void transactions.



Pay As You Go Delayed Authorization Model

American Express Authorization and Capture Options

You have two options for handling authorizations and captures.

Option 1:

1. Send an authorization request for a nominal amount.
2. During the travel period, collect the rider's tap data to calculate the aggregated fare.
3. When the floor limit is reached, send a capture to collect the accumulated amount.
4. After completing the capture, return to step 1 to handle subsequent trips and journeys.

Option 2:

1. Send an authorization request for a nominal amount.
2. During the travel period, collect the rider's tap data to calculate the aggregated fare.
3. If the number of trips exceeds the floor limit, send a delayed online authorization.
4. Upon completion of the delayed online authorization, return to step 1 to handle subsequent trips and journeys.

Implement the version that suits your business model. However, American Express requires your payment technology partner (PTP) to support nominal and full value authorizations during the L3 certification.

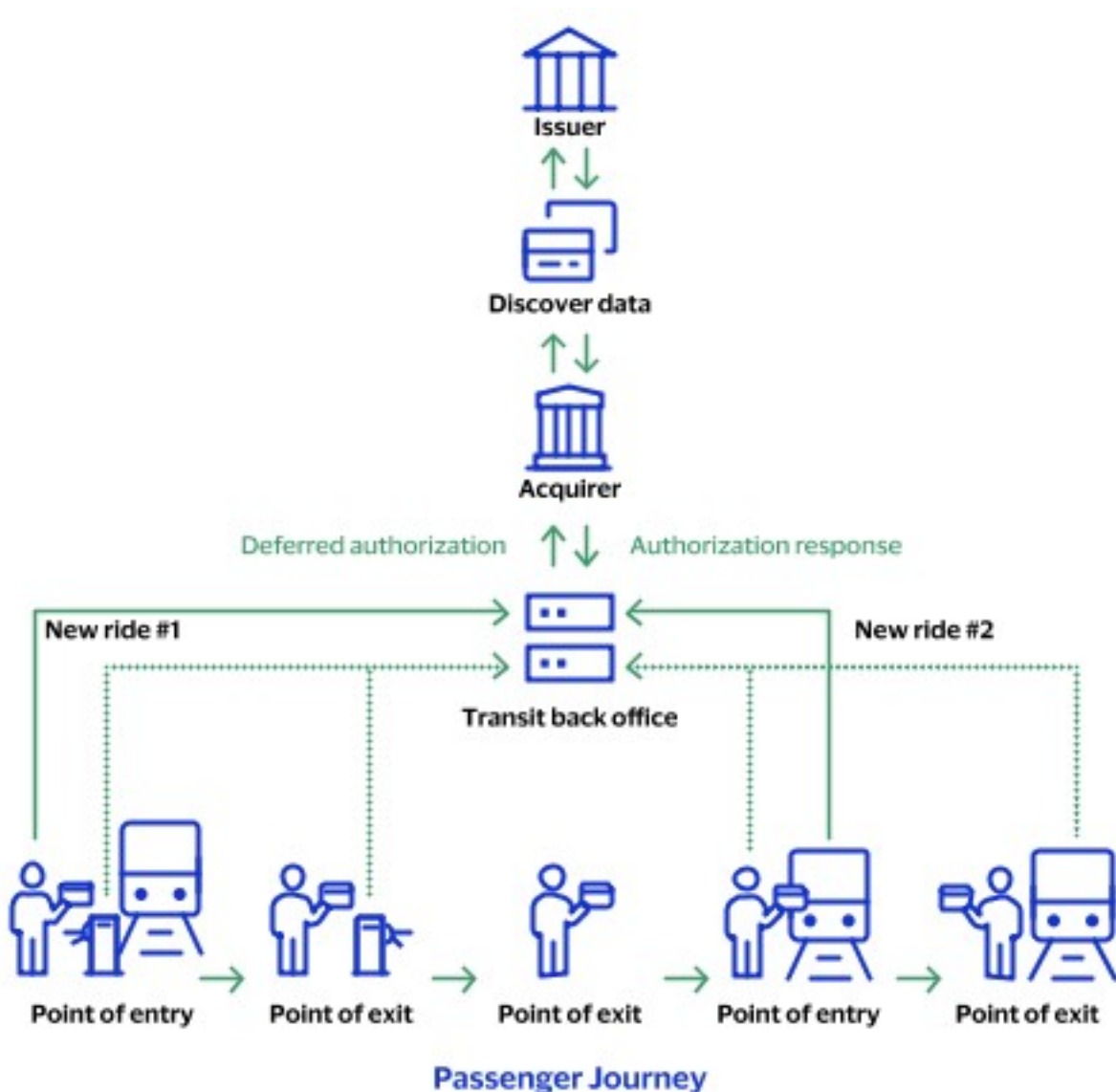
Discover Mass Transit Model

The Discover mass transit transaction model is Discover Pay As You Go (PAYG).

Discover Pay As You Go

These are the key characteristics of the Discover PAYG transaction model:

- Journeys are multimodal.
- Fares are based on distance.
- Points of entry are contactless.
- Accounts are verified with an authorization for a nominal amount or the maximum fare amount.
- The deny list is checked so that cardholders with previously declined transactions can be blocked.
- Data can be authenticated offline to confirm that the card is valid.
- Token Management Service (TMS) option for managing sensitive authentication data (SAD).
- Multiple card taps and fares can be aggregated into a single payment.
- The deny list is automatically updated every 20 minutes.
- The back office records all trips and fares in order to reconstruct journeys and calculate the final fare.
- When the nominal amount authorization is declined, debt recovery can be performed.
- Standard follow-on payment services can be used to capture, reverse, or void transactions.



Pay As You Go Model

Mastercard Mass Transit Model

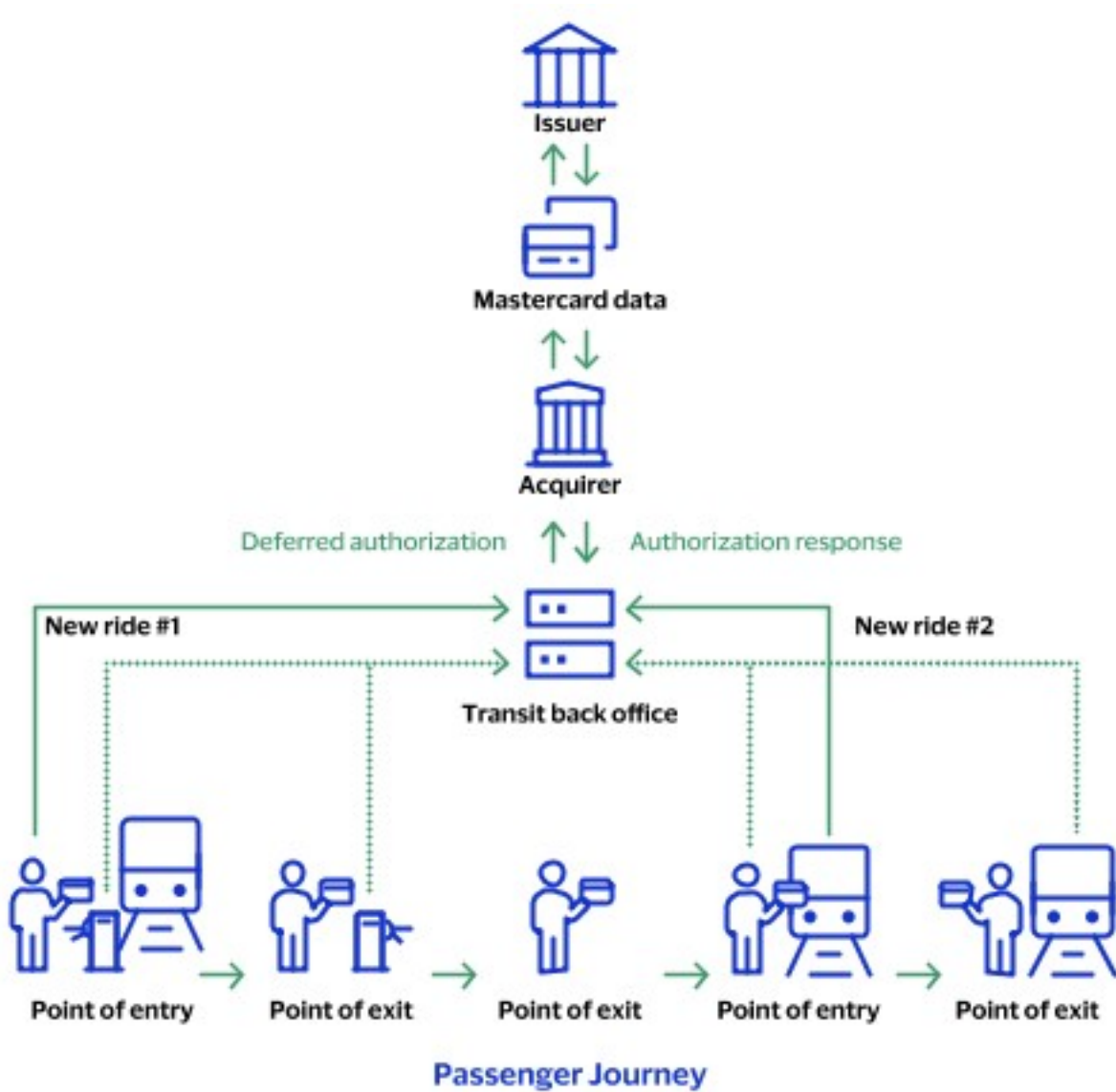
The Mastercard mass transit transaction model is Mastercard Pay As You Go. See [Mastercard Pay As You Go](#) on page 26.

Mastercard Pay As You Go

These are the key characteristics of the Mastercard Pay As You Go (PAYG) transaction model:

- Journeys are multimodal.
- Fares are based on distance.
- Points of entry are contactless.
- Accounts are verified with an authorization for a nominal amount or the maximum fare amount.

- The deny list is checked so that cardholders with previously declined transactions can be blocked.
- Data can be authenticated offline to confirm that the card is valid.
- Token Management Service (TMS) option for managing sensitive authentication data (SAD).
- Multiple card taps and fares can be aggregated into a single payment.
- The deny list is automatically updated every 20 minutes.
- The back office records all trips and fares in order to reconstruct journeys and calculate the final fare.
- When the nominal amount authorization is declined, debt recovery can be performed.
- Standard follow-on payment services can be used to capture, reverse, or void transactions.



Mastercard Pay As You Go Model

Visa Mass Transit Model

The Visa mass transit transaction model is Visa Mobility and Transport Transactions (MTT). See [Visa Mobility and Transport Transaction](#) on page 29.

Characteristics of the Visa MTT Model

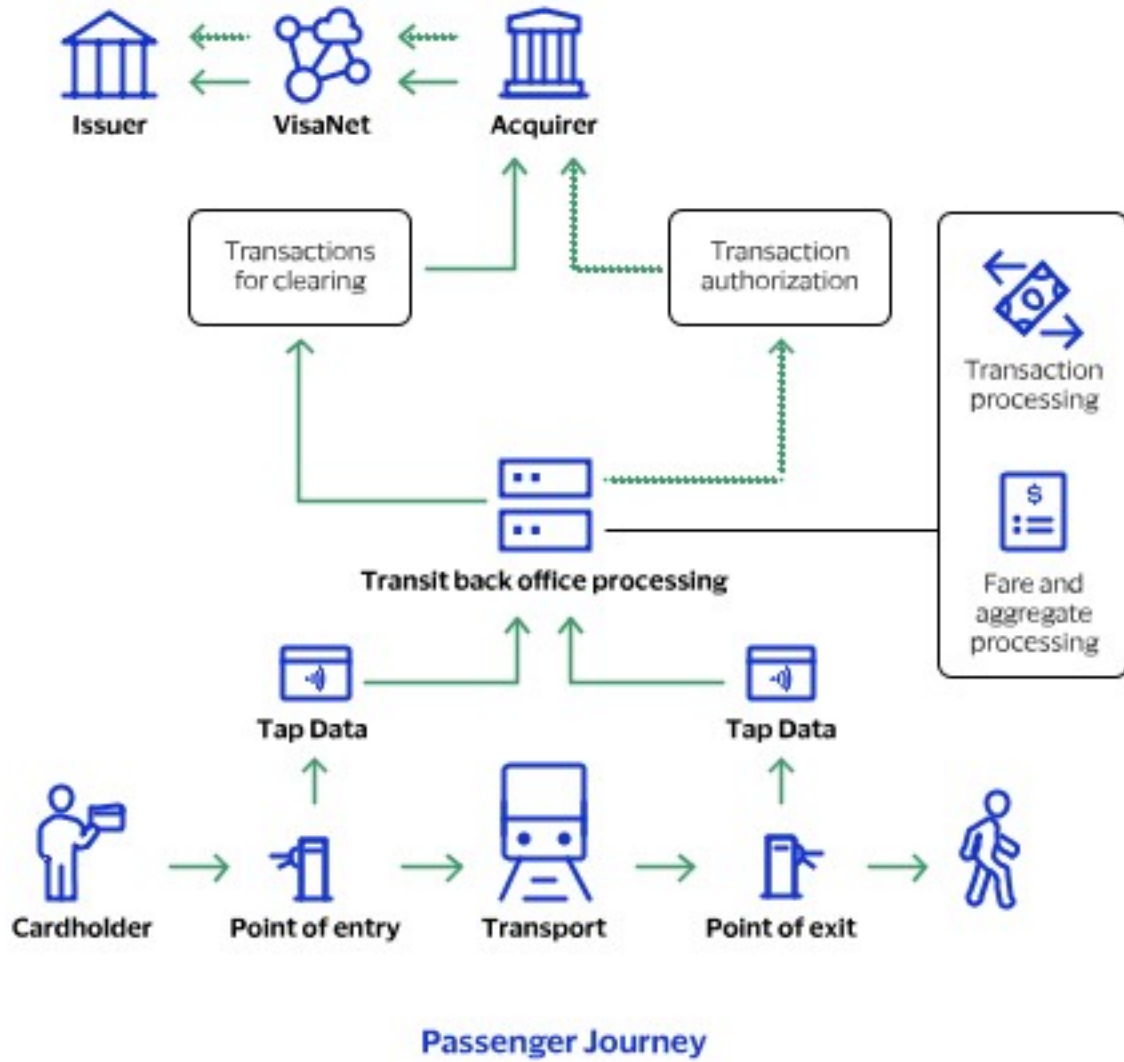
Characteristics	MTT
Designed for very high customer throughput.	Yes
Fare amount always known at the time the journey is started.	No
Transit reader accepts contactless payments only.	Yes

Characteristics	MTT
Intended for complex fares, including “capping” or multimodal.	Yes
Allows accumulation of multiple journeys into a single transaction.	Yes
Account verification requests (AVRs) performed on card’s first use.	Yes
Special liability model included (chargeback threshold).	Yes
Requires declined cards to be blocked using deny lists.	Yes
Requires merchant back office for fare calculation.	Yes
Intended authorization model.	Deferred
Authorization resubmissions for debt recovery.	Yes

Visa Mobility and Transport Transaction

These are the key characteristics of the Visa Mobility and Transport Transaction (MTT) model:

- Journeys are multimodal.
- Fares are based on distance.
- Points of entry are contactless.
- Accounts are verified with an account verification request (AVR) authorization for a nominal amount or the maximum fare amount.
- The deny list is checked so that cardholders with previously declined transactions can be blocked.
- Data can be authenticated offline to confirm the card is valid.
- Token Management Service (TMS) option for managing sensitive authentication data (SAD).
- Multiple card taps can be aggregated into a single payment.
- The back office manages the deny list and distributes them to terminals.
- The back office records all trips and fares to perform journey reconstruction to calculate the final fare.
- Standard follow-on payment services can be used to capture, reverse, or void transactions.
- First ride risk protection when the first authorization fails.
- When the AVR authorization is declined, debt recovery can be performed



Visa Mobility and Transport Transaction Model

Related concepts

- [Visa Mobility and Transport Transaction Workflow on page 36](#)

Mass Transit Transaction Workflows

Each card type has a unique transit transaction workflow.

American Express:

- [American Express Pay As You Go Workflow](#) on page 31

Discover workflow:

- [Discover Pay As You Go Workflow](#) on page 33

Mastercard workflow:

- [Mastercard Pay As You Go Workflow](#) on page 35

Visa workflow:

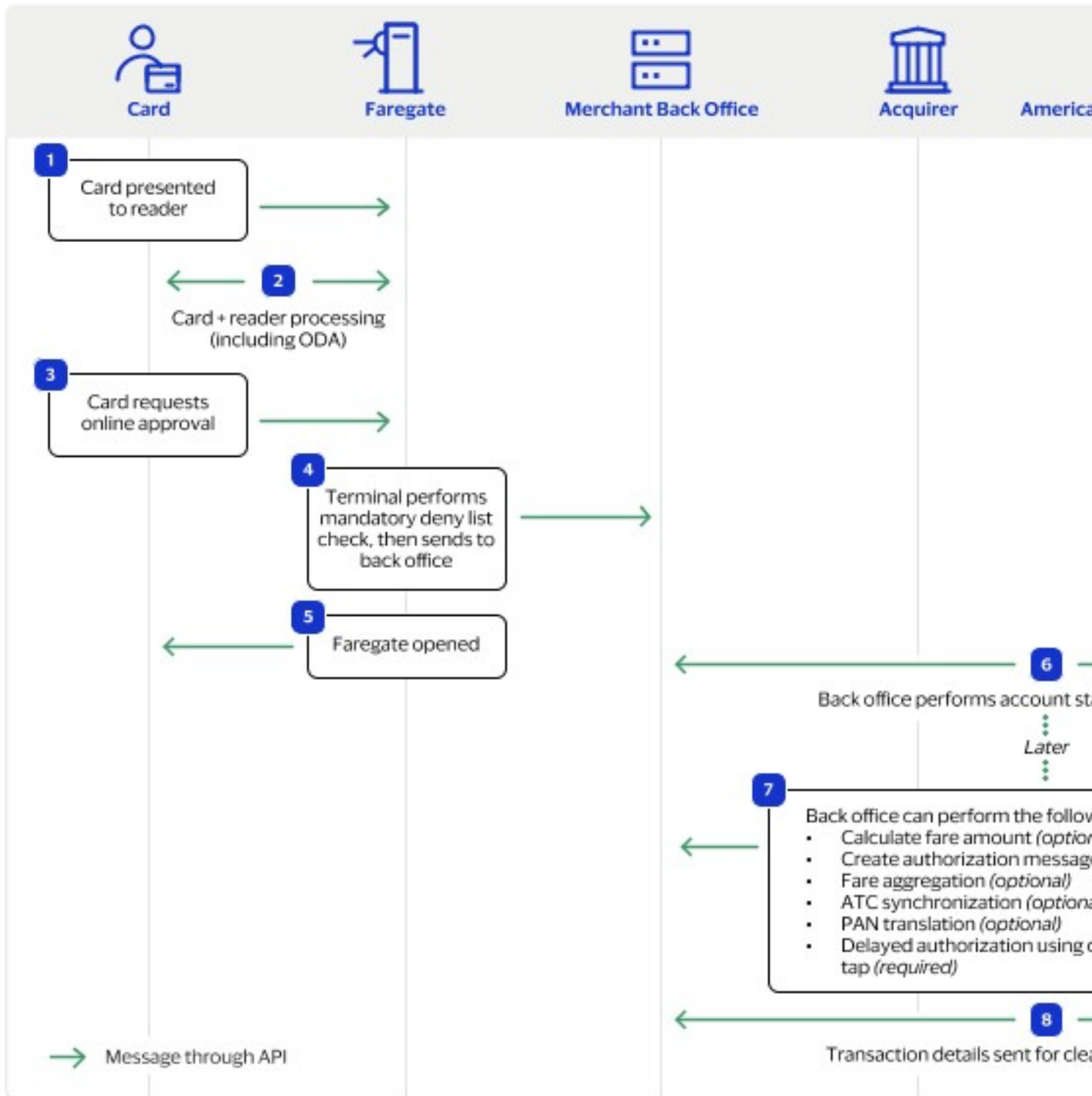
- [Visa Mobility and Transport Transaction Workflow](#) on page 36

Additional workflows:

- [Near Real-Time Workflow](#) on page 38
- [Fare Calculation and Submission Workflow](#) on page 40
- [Debt Recovery Workflows](#) on page 42

American Express Pay As You Go Workflow

The American Express Pay As You Go (PAYG) is a delayed authorization using the Expresspay transit policy workflow. It begins when the rider taps a payment card at the fare collection terminal.



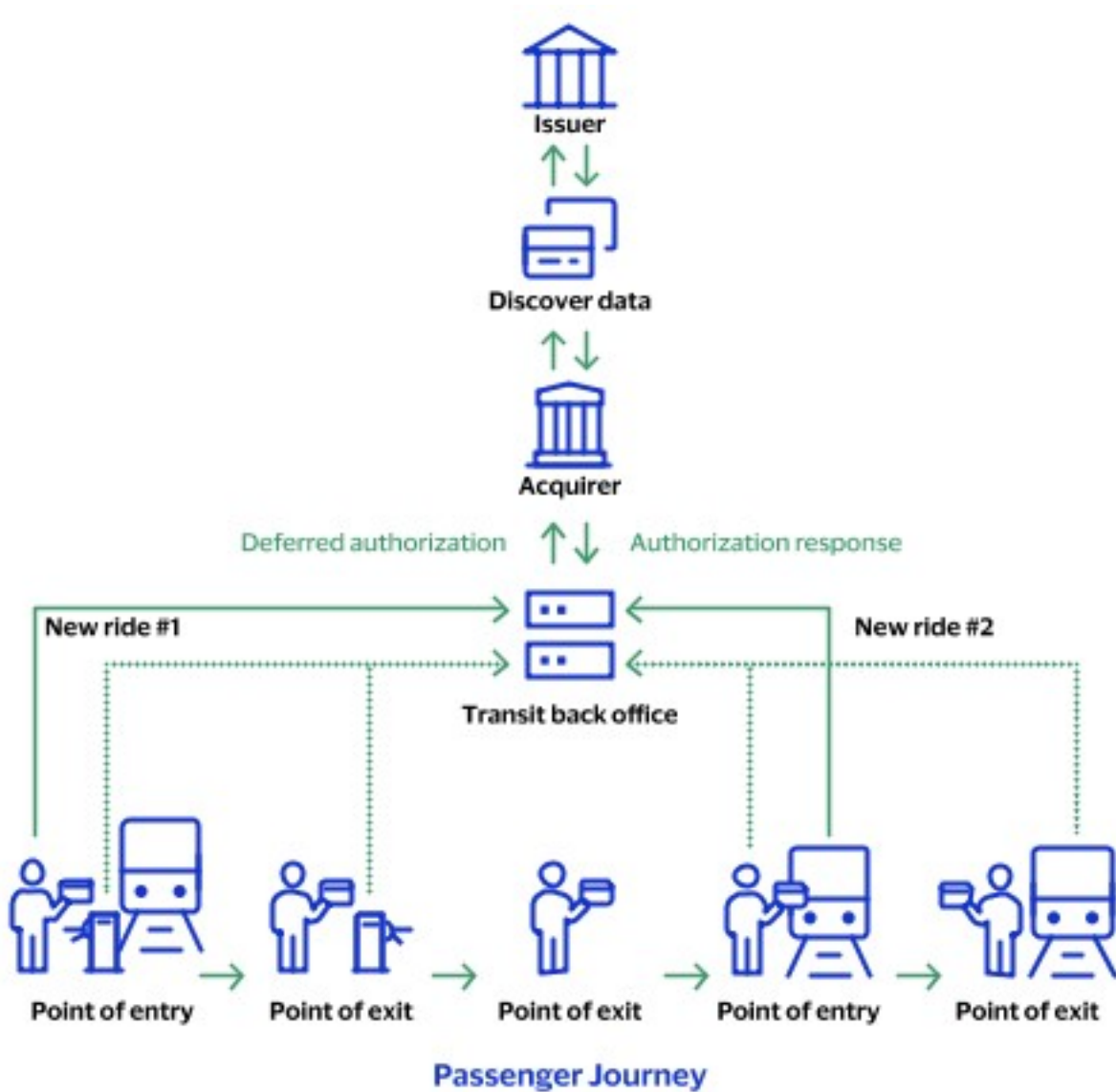
Pay As You Go Delayed Authorization Model

1. The cardholder taps the card to enter the transit system.

2. The gate validates the card using offline data authentication (ODA), the card expiration date, and the deny list.
3. When the card is valid, the gate allows the passenger to enter the transit system.
4. When the ODA fails, the card is added to the deny list, and the debt recovery process begins. See [Debt Recovery Workflows](#) on page 42.
5. You send an authorization request for a nominal amount. See [American Express Delayed Online Authorization with EMV Data](#) on page 58.
6. When the authorization is successful, you calculate the fare for the travel period.
7. At the last tap of the day, submit a follow-on capture request for the day's aggregated fare amount. For authorization and capture options, see [American Express Authorization and Capture Options](#) on page 24.

Discover Pay As You Go Workflow

The Discover Pay As You Go (PAYG) workflow begins when the rider taps a payment card at the fare collection terminal.

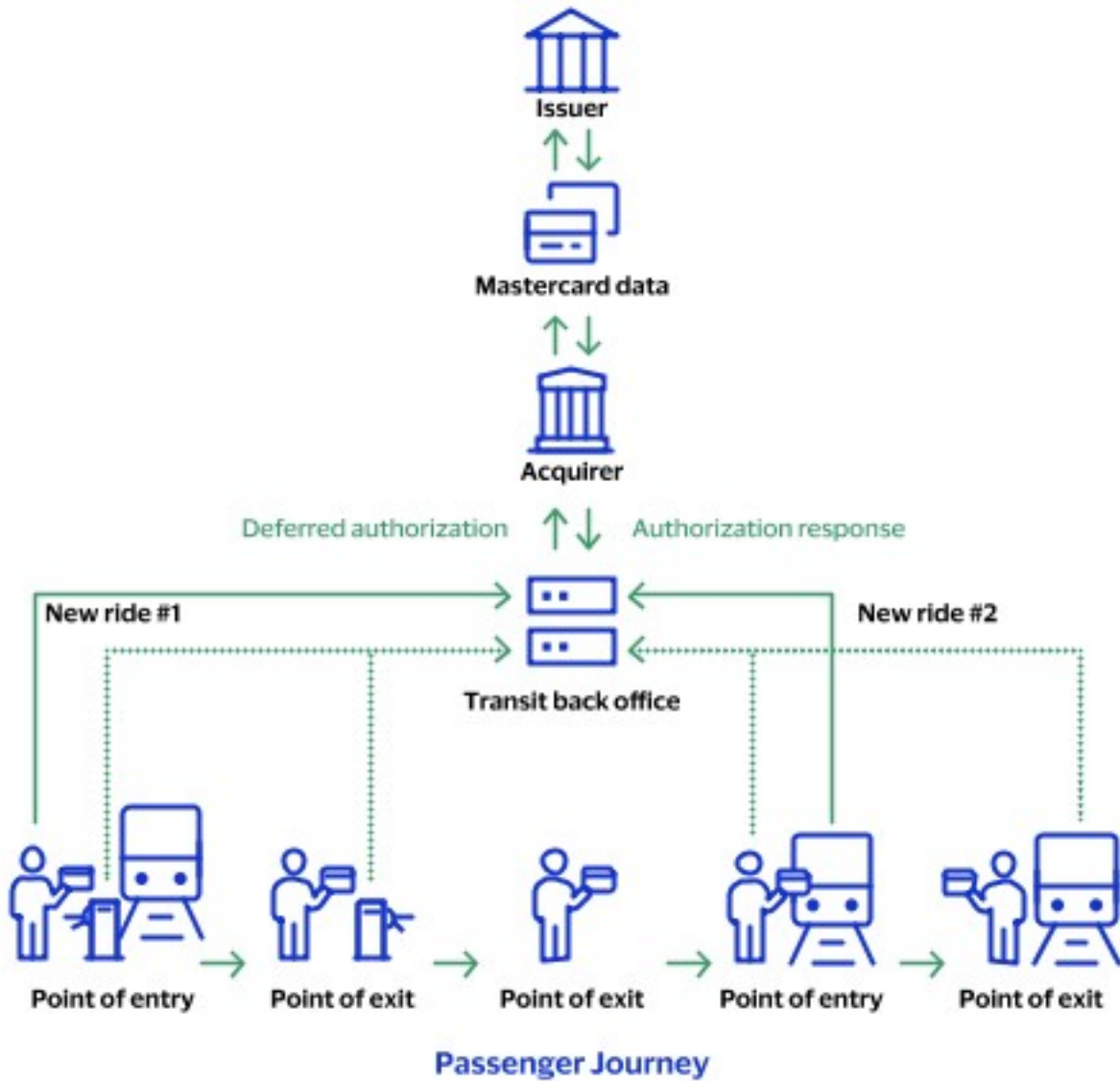


Pay As You Go Model

1. The cardholder taps the card to enter the transit system.
2. The gate validates the card using offline data authentication (ODA), the card expiration date, and the deny list.
3. When the card is valid, the gate allows the passenger to enter the transit system.
4. When the ODA fails, the card is added to the deny list, and the debt recovery process begins. See [Debt Recovery Workflows](#) on page 42.
5. You send an authorization request for a nominal amount. See [Discover Authorization with EMV Data](#) on page 61.
6. When the authorization is successful, you calculate the fare for the travel period.
7. When the fare is more than 15.00 USD, send an authorization or sale request for the higher amount. See [Discover Sale with EMV Data](#) on page 72.
8. At the last tap of the day, submit a follow-on capture request for the day's aggregated fare amount.

Mastercard Pay As You Go Workflow

The Mastercard Pay As You Go (PAYG) workflow begins when the rider taps a payment card at the fare collection terminal.



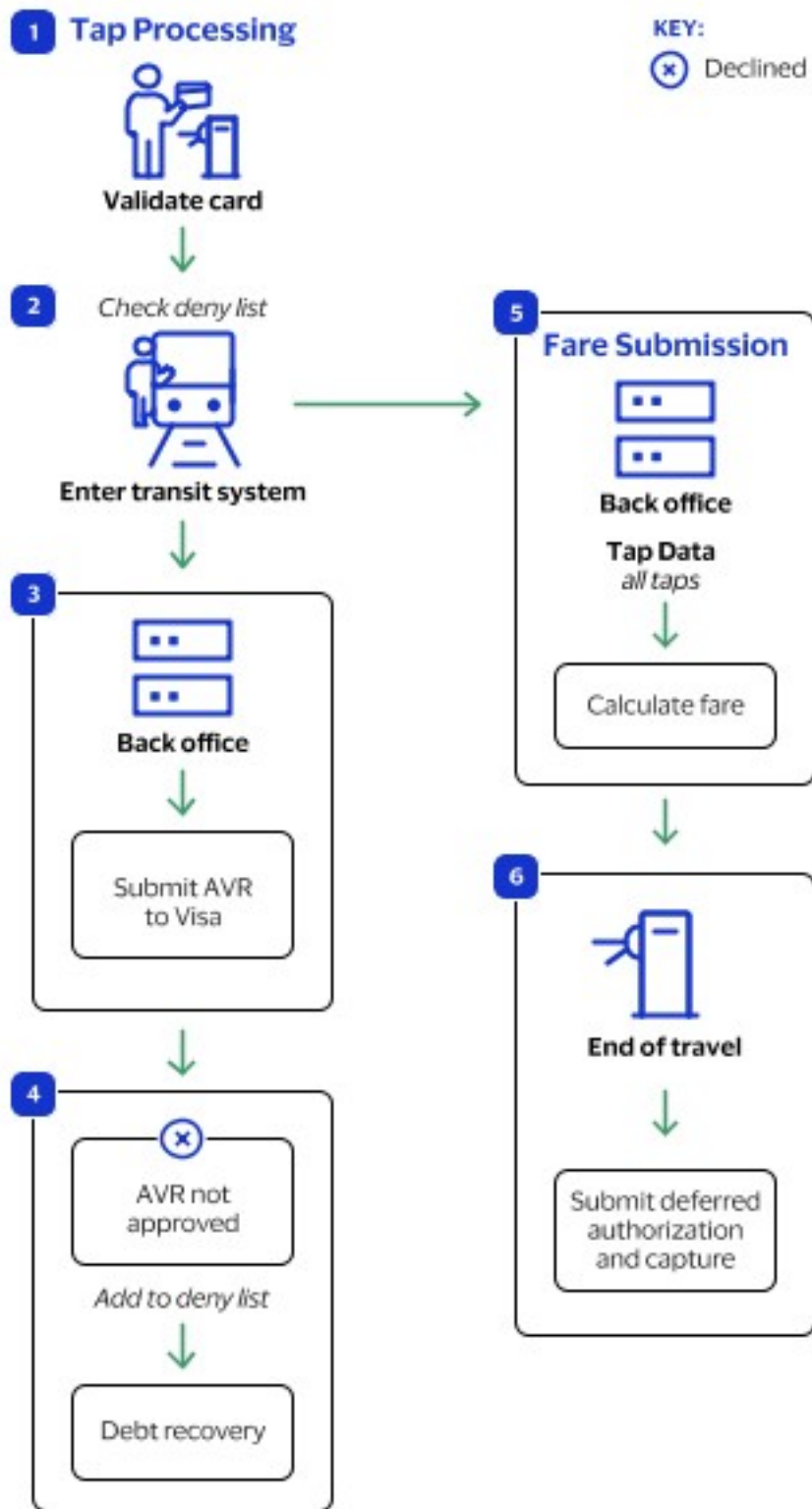
Pay As You Go Model

1. The cardholder taps the card to enter the transit system.
2. The gate validates the card using Mastercard combined data authentication (CDA), card expiration date, and the deny list.
3. When the card is valid, the gate allows the passenger to enter the transit system.
4. When the CDA fails, the card is added to the deny list, and the debt recovery process begins. See [Debt Recovery Workflows](#) on page 42.
5. You send an authorization request for a nominal amount. See [Mastercard Authorization with EMV Data](#) on page 64.

6. When the authorization is successful, you calculate the fare for the travel period and submit a sale request at the end of the travel period.

Visa Mobility and Transport Transaction Workflow

The Visa Mobility and Transport Transaction (MTT) workflow begins when the rider taps a payment card at the fare collection terminal.



Visa Mobility and Transport Transaction Workflow

1. The cardholder taps the card to enter the transit system.

2. The validator checks the deny list to determine card validity, and allows the rider to enter the transit system.
3. The back office submits an account verification request (AVR) to Cybersource. See [Visa Account Verification Request \(AVR\) with EMV Data](#) on page 67.
4. When the authorization fails, the card is added to the deny list, and the debt recovery process begins. See [Debt Recovery Workflows](#) on page 42.
5. During the travel period, the back office collects the rider's tap data to calculate the fare.
6. At the end of the travel period, the back office submits a deferred authorization and capture request. See the [Fare Calculation and Submission Workflow](#) on page 40.

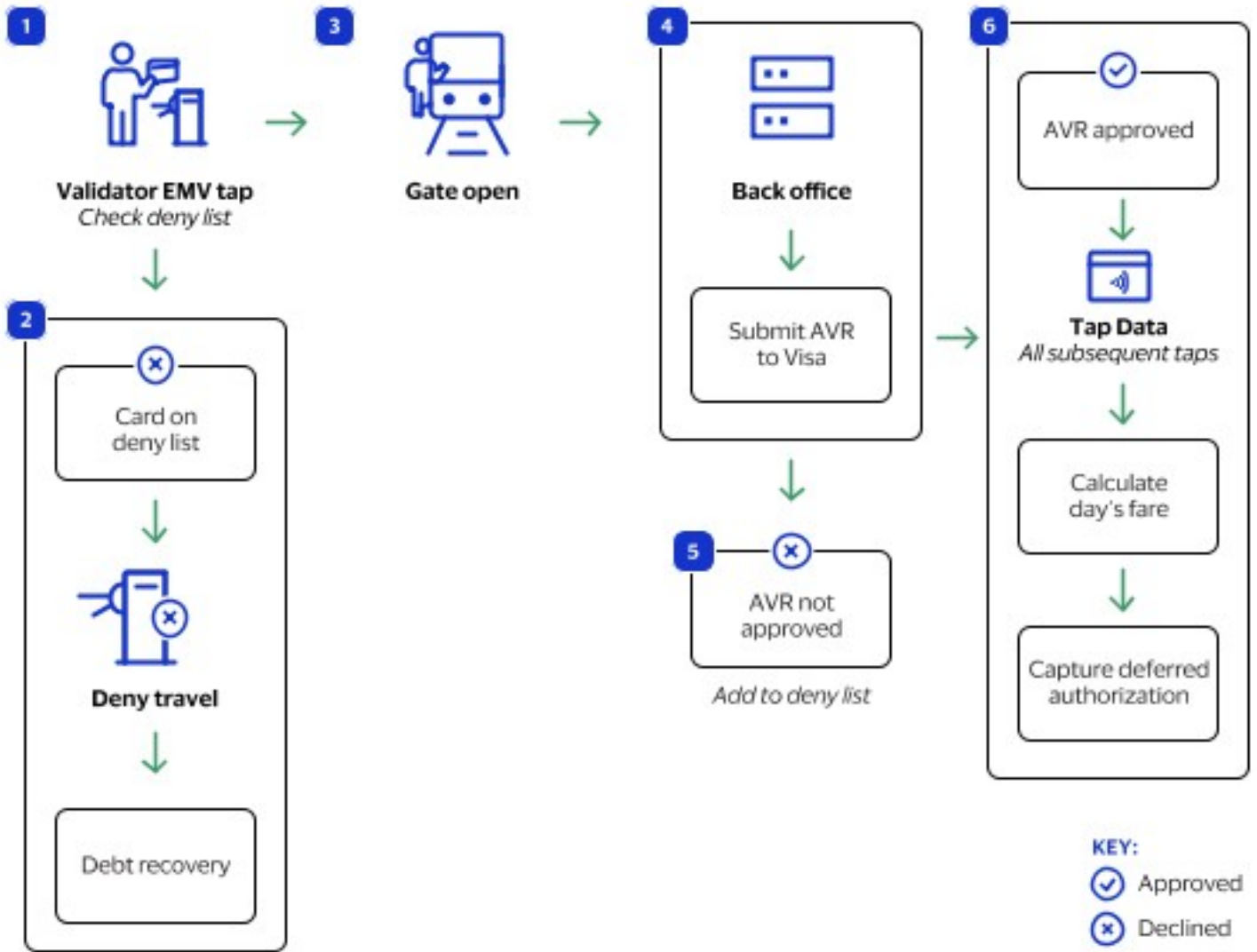
Additional Workflows

This section describes additional workflows that are not card-specific.

- [Near Real-Time Workflow](#) on page 38
- [Fare Calculation and Submission Workflow](#) on page 40
- [Debt Recovery Workflows](#) on page 42

Near Real-Time Workflow

The near real-time workflow begins when the cardholder taps a payment card at the validator to enter the transit system.

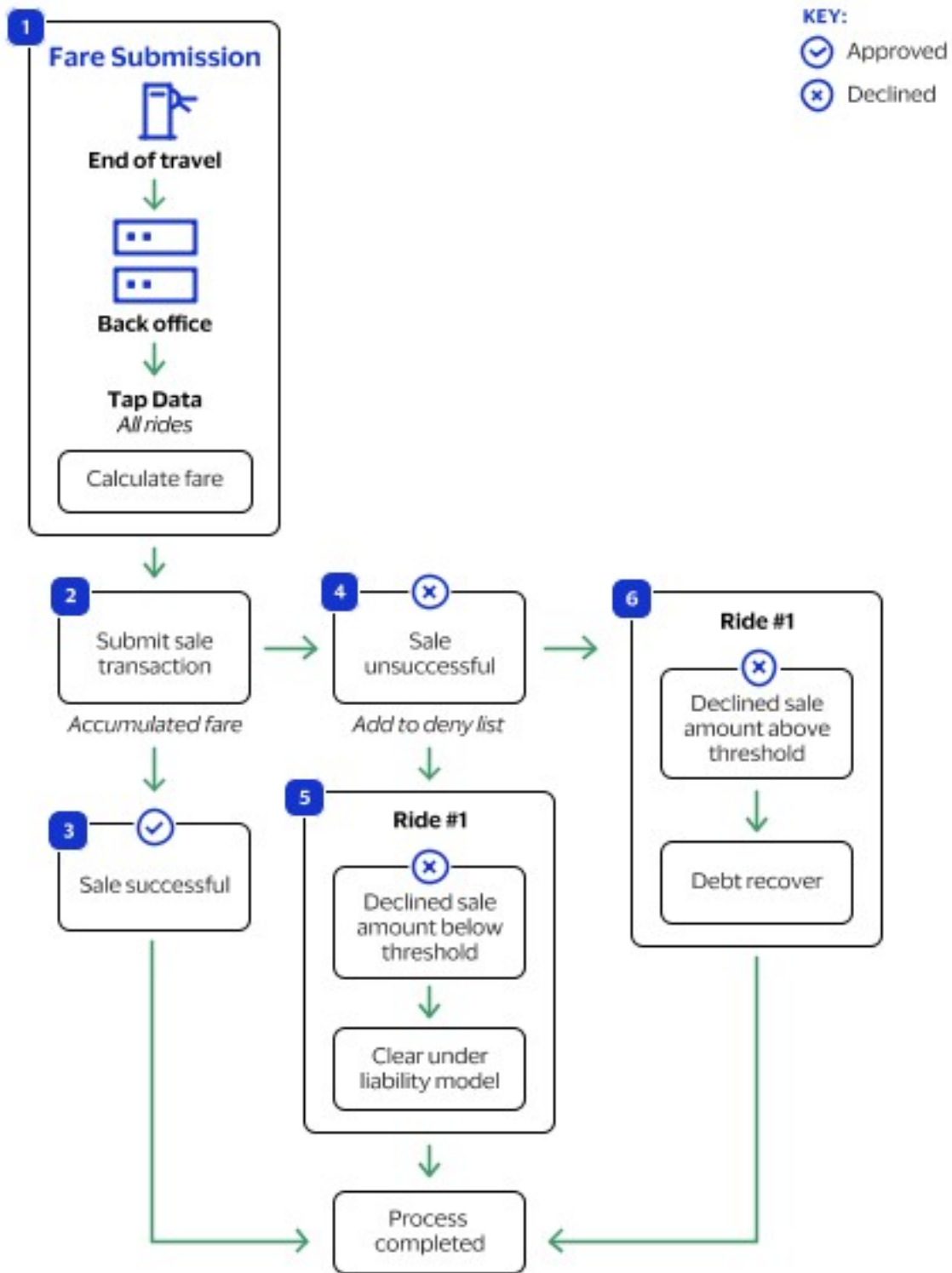


Near Real-Time Workflow

1. The validator checks the deny list for the payment card.
2. When the card is on the deny list due to a previous failed payment, the validator does not open the gate, and the payment is processed through a debt recovery workflow. See the [Debt Recovery Workflows](#) on page 42.
3. When the card is not on the deny list, the validator opens the transit gate for the cardholder to travel.
4. A new transient token is generated for processing the account verification request (AVR).
5. The back office sends an account verification request (AVR) to Cybersource.
6. When the AVR fails, the card is added to the deny list and might be eligible for the first ride risk debt recovery.
7. When the AVR is successful, the card data is used to track subsequent taps, calculate fares for the day, and capture the deferred authorization.

Fare Calculation and Submission Workflow

The fare calculation workflow begins at the end of the travel period.



Visa Fare Calculation and Submission Flow

1. The back office calculates the fare of all rides taken during the travel period.

2. You request a sale transaction for the accumulated fare. See [Visa Deferred Sale with EMV Data](#) on page 76.
3. When the sale is successful, the process is complete.
4. When the sale is declined, the card hash is added to the deny list.
5. When the declined sale amount is above the chargeback threshold, the transaction is moved to debt recovery. See [Debt Recovery Workflows](#) on page 42.
6. When the declined sale amount is for a first ride and below the chargeback threshold, you can request the payment using the first ride risk chargeback rules as defined by each card scheme.

Debt Recovery Workflows

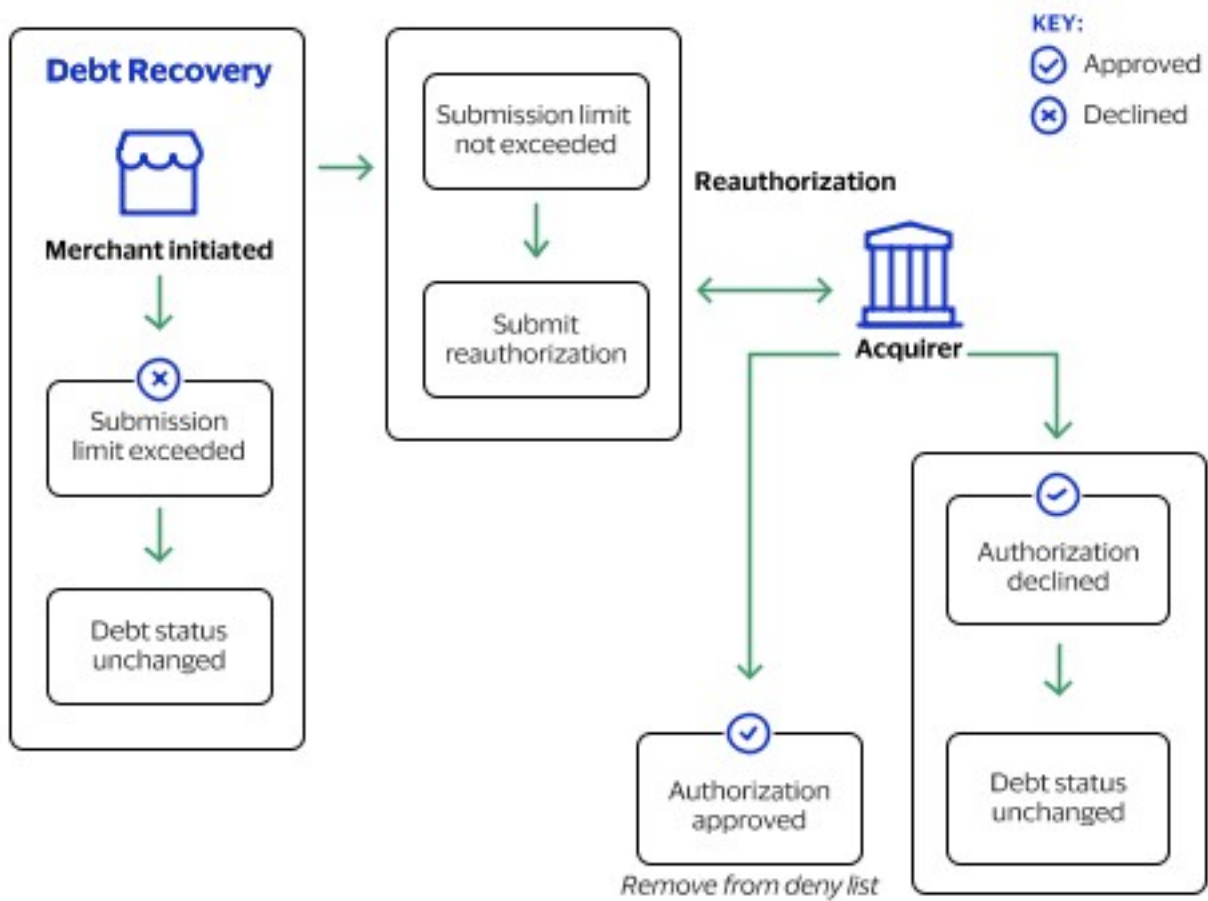
A debt recovery transaction is required in order to retrieve outstanding debt in the event of a decline of the end-of-day transaction. A debt recovery transaction is also required in order to remove a card from a deny list. You must remove the card from the deny list within one hour of receiving the authorization approval. These are the ways to recover debt:

- Merchant-initiated transaction, resubmitted using the card number.
- Tap-initiated transaction using the EMV track 2 equivalent and EMV tags created when the cardholder returns to enter the transit system.
- Cardholder-initiated transaction through an e-commerce website or by a phone call.

When the debt recovery transaction is declined, you can request payment using the first ride risk liability model. See the card scheme rules for mass transit transaction chargebacks.

Merchant-Initiated Transaction Resubmission

A merchant-initiated transaction (MIT) resubmission is a deferred authorization that originates from your back office. This transaction typically references the original declined end-of-day transaction and uses the card number. Visa allows up to six authorization resubmissions within 14 days.



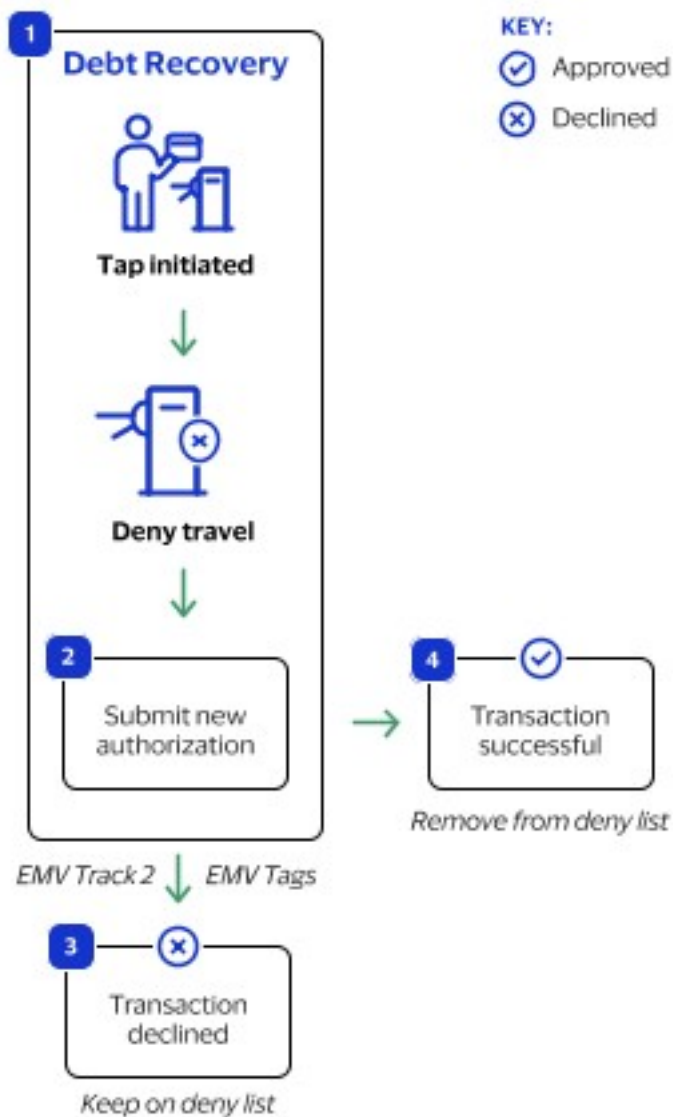
Merchant-Initiated Visa Debt Recovery Flow

1. When the amount exceeds the debt recovery amount limit, the card remains on the deny list.
2. When the amount is below the debt recovery amount limit, send a sale request. See [Merchant-Initiated Sale for Debt Recovery with Stored Card Data](#) on page 96.
3. When the transaction is declined, keep the card on the deny list.
4. When the transaction is successful, remove the card from the deny list.

Tap-Initiated Debt Recovery

Tap-initiated debt recovery occurs when the cardholder returns to the transit gate and the validator recognizes a new contactless tap.

You may deny the rider entrance unless the tap-initiated debt recovery is attempted in real time while the cardholder is at the gate. The authorization request uses the EMV track 2 equivalent and EMV tags from the new tap and includes a future capture date.



Cardholder-Initiated Debt Recovery Flow

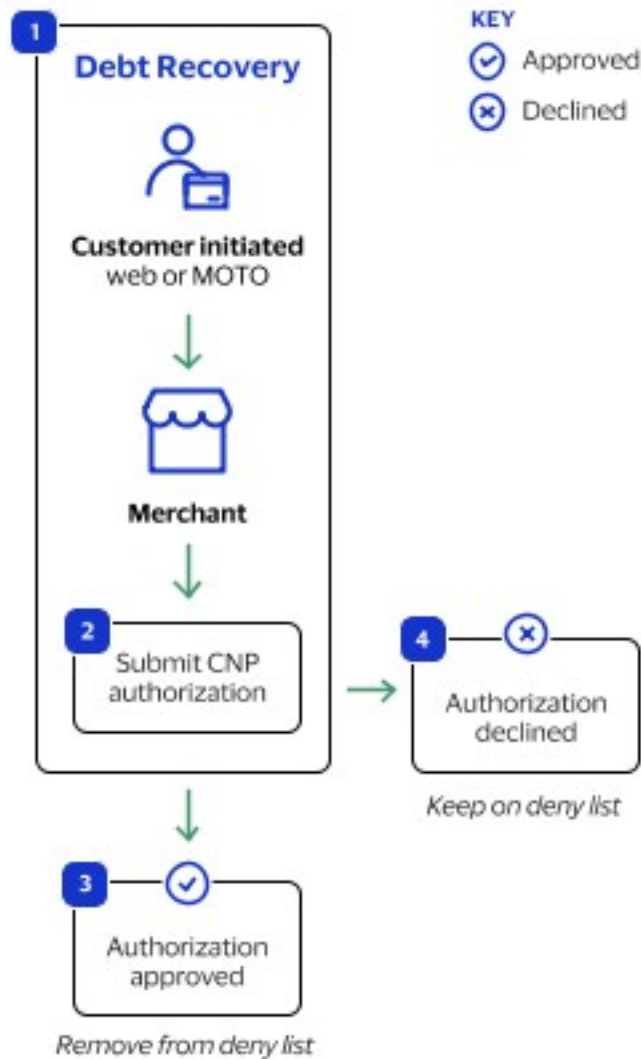
1. Cardholder taps their card to enter the transit system.
2. You submit a new authorization request using the EMV track 2 equivalent and EMV tags created by the validator and a capture date in the future.
3. When the transaction is declined, keep the card on the deny list.
4. When the transaction is successful, remove the card from the deny list.

Cardholder-Initiated Debt Recovery

These are the scenarios for a cardholder-initiated debt recovery:

- E-commerce, at your website.
- Mail order or telephone order (MOTO), on a phone call.

For e-commerce and MOTO payment services, see the [Payments Developer Guide](#).



Cardholder-Initiated Debt Recovery Flow

1. Cardholder contacts you through your website or by phone call.
2. You process a card-not-present authorization.
3. When the request is successful, remove the card from the deny list.
4. When the request fails, leave the card on the deny list.

Mass Transit Payment Services Using EMV and Card Data

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

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

This table shows which EMV tags are:

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

EMV Data Elements and Tags

Data Element	EMV Tag	American Express	Discover PAYG	Mastercard PAYG	Visa MTT
Transaction Date	9A	M	M	M	M
Transaction Type	9C	M	M	M	M
Transaction Currency Code	5F2A	M	M	M	M
Terminal Country Code	9F1A	M	M	M	M
Amount Authorized	9F02	M	M	M	M
Amount Other	9F03	M	M	M	M
Application PAN Sequence Number	5F34	M	P	C	O

Data Element	EMV Tag	American Express	Discover PAYG	Mastercard PAYG	Visa MTT
Application Transaction Counter (ATC)	9F36	M	M	M	M
Application Interchange Profile (AIP)	82	M	M	M	M
Dedicated File (DF) Name	84	M	M	M	M
Terminal Verification Results (TVR)	95	M	M	M	M
Issuer Application Data	9F10	M	M	M	M
Application Cryptogram	9F26	M	M	M	M
Cryptogram Information Data (CID)	9F27	M	O	M	O
Terminal Capabilities	9F33	M	M	M	M
Cardholder Verification Method (CVM) Results	9F34	O	O	M	O
Unpredictable Number (UN)	9F37	M	M	M	M
Form Factor Indicator	9F6E	C*	C	O (Authorizations) P (Refunds)	C
Mastercard Authenticated Application Data	9F60	Does not apply	Does not apply	O	Does not apply
Mastercard Kernel Identifier#Terminal	96	Does not apply	Does not apply	O	Does not apply

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

Transaction Types

When you include the transaction type in your request, the description appears in the Business Center and transaction reports.

Transactions are grouped into these categories:

- **TransitDA:** deferred-aggregated (DA) transactions, which are also known as Visa MTT and Mastercard PAYG transactions.
- **BAU:** business-as-usual transactions that represent no exceptions or errors for cardholders.
- **FRR:** first-ride-risk transactions that occur where first-ride-risk liability shift is being operated. These are specific to a card scheme and region.
- **DR:** debt-recovery transactions initiated by the merchant or when the cardholder taps a contactless card at a validator to enter the transit system.
- **DR CIT:** debt recovery customer-initiated transactions initiated by the cardholder when they explicitly pay a debt, including e-commerce and telephone orders.
- **Service:** standard transactions for completing a payment.
- **Error:** standard transactions for handling transaction errors.

To include the transaction type, set the **clientReferenceInformation.comments** request field to the transaction value corresponding to the service description. These tables provide the value for each type of transaction:

Business as Usual (BAU)

Service	Field Value	Description
authorization	TransitDA BAU zero value auth	Zero amount authorization to verify a card.
authorization	TransitDA BAU nominal value auth	Nominal value authorization to verify a card.
authorization	TransitDA BAU full value auth	Deferred aggregated authorization for the aggregated value that is sent at the end of the travel period.
sale	TransitDA BAU full value sale	Deferred aggregated authorization and capture for the aggregated value that is sent at the end of the travel period.
capture	TransitDA BAU capture	Capture of any business as usual authorization. Could be a nominal authorization or full value authorization.

Service	Field Value	Description
capture	TransitDA BAU capture (split)	Capture without a previous authorization. Used by Mastercard PAYG in the UK.
authorization	TransitDA BAU registration auth	Zero amount authorization as part of journey history service. Could include CVV2 and 3-D Secure 2.x.

First Ride Risk (FRR)

Service	Field Value	Description
authorization	TransitDA FRR full auth	Full amount authorization for a previous verification authorization request that was declined. Decline response is common.
capture	TransitDA FRR capture	Forced capture of a declined authorization when FRR funding applies.
authorization	TransitDA FRR MIT DR auth	Merchant-initiated authorization to clear a debt status after the TransitDA FRR authorization is processed. If successful, the FRR capture is reversed.
reversal	TransitDA FRR MIT DR reversal	Reversal sent if previous TransitDA FRR MIT DR auth was successful.
authorization	TransitDA FRR tap DR auth	Authorization sent following a card tap to clear a debt status after TransitDA FRR authorization is processed. If successful, the FRR capture is reversed.
reversal	TransitDA FRR tap DR reversal	Reversal when a TransitDA FRR tap DR auth was successful.

Debt Recovery (DR)

Service	Field Value	Description
sale	TransitDA Debt recovery MIT sale F PAN TransitDA Debt recovery MIT sale DPAN	Merchant-initiated debt recovery authorization and capture using a FPAN or DPAN.
authorization	TransitDA Debt recovery MIT auth FPAN TransitDA Debt recovery MIT auth DPAN	Merchant-initiated debt recovery authorization using a FPAN or DPAN.
capture	TransitDA Debt recovery MIT capture	Merchant-initiated debt recovery capture of a previous TransitDA Debt recovery MIT auth transaction.
sale	TransitDA Debt recovery tap sale	Tap-initiated EMV debt recovery authorization and capture.
authorization	TransitDA Debt recovery tap auth	Tap-initiated EMV debt recovery authorization.
capture	TransitDA Debt recovery tap capture	Tap-initiated EMV debt recovery capture of a previous TransitDA Debt recovery tap auth transaction.

Cardholder-Initiated Debt Recovery (DR CIT)

Service	Field Value	Description
sale	TransitDA Debt recovery CIT Ecom sale	Cardholder-initiated debt recovery authorization and capture.
authorization	TransitDA Debt recovery CIT Ecom auth	Cardholder-initiated debt recovery authorization.
capture	TransitDA Debt recovery CIT Ecom capture	Cardholder-initiated debt recovery capture of a previous TransitDA Debt recovery CIT Ecom auth transaction.
sale	TransitDA Debt recovery CIT Ecom 3DS 2 sale	Cardholder-initiated debt recovery authorization and capture.
authorization	TransitDA Debt recovery CIT Ecom 3DS 2 auth	Cardholder-initiated debt recovery authorization.

Service	Field Value	Description
capture	TransitDA Debt recovery CIT Ecom 3DS 2 capture	Cardholder-initiated debt recovery capture of a previous TransitDA Debt recovery CIT Ecom 3DS2 auth transaction.
sale	TransitDA Debt recovery CIT Moto sale	Cardholder-initiated debt recovery authorization and capture.
authorization	TransitDA Debt recovery CIT Moto auth	Cardholder-initiated debt recovery authorization.
capture	TransitDA Debt recovery CIT Moto capture	Cardholder-initiated debt recovery capture of previous TransitDA Debt recovery CIT Moto auth transaction.

Services

Service	Field Value	Description
refund	REFUND Automatic	Programmatic follow-on refund for a previous capture.
credit	CREDIT Automatic	Programmatic stand-alone credit.
refund	REFUND Manual	Manual follow-on refund for a previous capture.
credit	CREDIT Manual	Manual stand-alone credit.

Errors

Service	Field Value	Description
reversal	REVERSAL Timeout	Reversal of a previous request for which a response was not received.
reversal	REVERSAL other	Reversal for an authorization for a reason other than TransitDA FRR MIT DR reversal, TransitDA FRR tap DR reversal, or REVERSAL Timeout.
void	VOID Timeout	Void of a previous request for which a response was not received.
void	VOID Payment	Void of a payment within the same day.
void	VOID Capture	Void of a capture within the same day.

Service	Field Value	Description
void	VOID Refund	Void of a refund within the same day.
void	VOID Credit	Void of a credit within the same day.

Required Fields for Authorizations with EMV Data

This table provides information about the fields required to process authorizations with EMV data.

Authorizations with EMV Data Fields

REST API Field	American Express Account Status Check	American Express Delayed Online	Discover PAYG	Mastercard PAYG	Visa AVR	Information/Value
clientReferenceInformation.code						—
clientReferenceInformation.comments						For this value, see Transaction Types on page 48.
clientReferenceInformation.partner.solutionId						Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber						Cybersource provides the value for this field.
clientReferenceInformation.transactionId						—
orderInformation.amountDetails.currency						—
orderInformation.amountDetails.totalAmount						For Visa AVR, set this field to 0.00.

REST API Field	American Express Account Status Check	American Express Delayed Online	Discover PAYG	Mastercard PAYG	Visa AVR	Information/Value
paymentInformation.card.type						—
paymentInformation.initiationChannel	—	—	—		—	—
pointOfSaleInformation.catLevel						Set this field to 2.
pointOfSaleInformation.emv.cardSequenceNumber				—		For Visa, set this field to 1.
pointOfSaleInformation.emv.tags						—
pointOfSaleInformation.entryMode						Set this field to contactless.
pointOfSaleInformation.serviceCode	—	—	—		—	—
pointOfSaleInformation.terminalCapability						Set this field to 5.
pointOfSaleInformation.terminalId						—
pointOfSaleInformation.terminalPinCapability						Set this field to 0.
pointOfSaleInformation.trackData						—
processingInformation.authorizationOptions.aggregatedAuthIndicator		—			—	Set this field to true.
processingInformation.authorizationOptions.authIndicator	—	—	—		—	Set this field to 0.
processingInformation.authorizationOptions.deferredAuthIndicator		—			—	Set this field to true.

REST API Field	American Express Account Status Check	American Express Delayed Online	Discover PAYG	Mastercard PAYG	Visa AVR	Information/Value
processingInformation.authorizationOptions.transportationMode	—	—	—		—	—
processingInformation.captureOptions.dateToCapture			—		—	—
processingInformation.commerceIndicator						Set this field to retail.
processingInformation.industryDataType					—	Set this field to transit.

American Express Account Status Check Authorization with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#)

Set this field to `TransitDA BAU nominal value auth.`

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyAuthorizationId](#) Cybersource provides the value for this field.

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

Related information

- [API Field Reference for the REST API](#)

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

Request

```
{
  "orderInformation": {
    "amountDetails": {
      "currency": "EUR",
      "totalAmount": "3.00"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "003"
    }
  },
  "processingInformation": {
    "capture": false,
    "captureOptions": {
      "dateToCapture": "0901"
    }
  }
}
```

```

    },
    "industryDataType": "transit",
    "commerceIndicator": "retail",
    "authorizationOptions": {
      "partialAuthIndicator": false,
      "deferredAuthIndicator": true,
      "aggregatedAuthIndicator": true
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"9A032309019C01005F2A0209789F1A0203809F020600000000000009F030600000000000009F36020002820219C08408A00
      "cardSequenceNumber": "00"
    },
    "catLevel": "2",
    "entryMode": "contactless",
    "trackData": ";374245XXXXXXXXXX=241270115041234500000?",
    "terminalId": "12345678",
    "terminalCapability": "5",
    "terminalPinCapability": "0"
  },
  "clientReferenceInformation": {
    "comments": "TransitDA BAU nominal value auth",
    "code": "v7qWAIW6e",
    "partner": {
      "solutionId": "BUALWMZK",
      "thirdPartyCertificationNumber": "condue211609"
    },
    "transactionId": "Fg1xkLJGMmmmvwbB9qWAIW6e"
  }
}

```

Response to a Successful Request

```

{
  "_links":#{
    "authReversal":#{
      "method":#"POST",
      "href":#"/pts/v2/payments/6984001952686181104951/reversals"
    },
    "self":#{
      "method":#"GET",
      "href":#"/pts/v2/payments/6984001952686181104951"
    },
    "capture":#{
      "method":#"POST",
      "href":#"/pts/v2/payments/6984001952686181104951/captures"
    }
  },
  "clientReferenceInformation":#{
    "code":#"v7qWAIW6e",
    "partner":#{
      "solutionId":#"BUALWMZK"
    },
    "transactionId":#"Fg1xkLJGMmmmvwbB9qWAIW6e"
  }
}

```

```

},
"id":#"6984001952686181104951",
"orderInformation":#{
  "amountDetails":#{
    "authorizedAmount":#"3.00",
    "currency":#"EUR"
  }
},
"paymentAccountInformation":#{
  "card":#{
    "type":#"003"
  }
},
"paymentInformation":#{
  "accountFeatures":#{
    "category":#"AX",
    "group":#"0"
  },
  "tokenizedCard":#{
    "type":#"003"
  },
  "card":#{
    "type":#"003"
  }
},
"pointOfSaleInformation":#{
  "emv":#{
    "tags":#"9F2701809F34031F02025F340100"
  }
},
"processorInformation":#{
  "systemTraceAuditNumber":#"037806",
  "approvalCode":#"845614",
  "networktransactionId":#"001032401292273",
  "retrievalReferenceNumber":#"330009037806",
  "transactionId":#"001032401292273",
  "responseCode":#"00",
  "avs":#{
    "code":#"2"
  }
},
"reconciliationId":#"6984001952686181104951",
"status":#"AUTHORIZED",
"submitTimeUtc":#"2023-10-27T09:49:56Z"
}

```

American Express Delayed Online Authorization with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#)

Set this field to `TransitDA BAU nominal value auth.`

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyReferenceId](#) Cybersource provides the value for this field.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.type](#)

Set this field to `003`.

[pointOfSaleInformation.catLevel](#)

Set this field to `2`.

[pointOfSaleInformation.emv.cardSequenceNumber](#) Set this field to `00`.

[pointOfSaleInformation.emv.tags](#)

[pointOfSaleInformation.entryMode](#)

Set this field to `contactless`.

[pointOfSaleInformation.terminalCapability](#)

Set this field to `5`.

[pointOfSaleInformation.terminalId](#)

[pointOfSaleInformation.terminalPinCapability](#) Set this field to `0`.

[pointOfSaleInformation.trackData](#)

[processingInformation.captureOptions.dateToCapture](#)

[processingInformation.commerceIndicator](#) Set this field to `retail`.

processingInformation.industryDataTypeSet this field to `transit`.

Related information

- [API Field Reference for the REST API](#)

REST Example: American Express Delayed Online Authorization with EMV Data

Request

```
{
  "orderInformation": {
    "amountDetails": {
      "currency": "EUR",
      "totalAmount": "8.00"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "003"
    }
  },
  "processingInformation": {
    "captureOptions": {
      "dateToCapture": "0901"
    },
    "industryDataType": "transit",
    "commerceIndicator": "retail"
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"9A032309019C01005F2A0209789F1A0203809F020600000000000009F030600000000000009F36020002820219C08408A00"
      "cardSequenceNumber": "00"
    },
    "catLevel": "2",
    "entryMode": "contactless",
    "trackData": ";341111XXXXXXXXXX=241270215041234500000?";
    "terminalId": "12345678",
    "terminalCapability": "5",
    "terminalPinCapability": "0"
  },
  "clientReferenceInformation": {
    "comments": "TransitDA BAU full value auth",
    "code": "v7qWAIW6e",
    "partner": {
      "solutionId": "BUALWMZK",
      "thirdPartyCertificationNumber": "condue211609"
    },
    "transactionId": "Fg1xkLJGMmmmwB9qWAIW6e"
  }
}
```

Response to a Successful Request

```

{
  "_links":#{
    "authReversal":#{
      "method":#"POST",
      "href":#/pts/v2/payments/6984003567376178404953/reversals"
    },
    "self":#{
      "method":#"GET",
      "href":#/pts/v2/payments/6984003567376178404953"
    },
    "capture":#{
      "method":#"POST",
      "href":#/pts/v2/payments/6984003567376178404953/captures"
    }
  },
  "clientReferenceInformation":#{
    "code":#"v7qWAIW6e",
    "partner":#{
      "solutionId":#"BUALWMZK"
    },
    "transactionId":#"Fs8xkJGNslmvwbZ9qWAIW6e"
  },
  "id":#"6984003567376178404953",
  "orderInformation":#{
    "amountDetails":#{
      "authorizedAmount":#"8.00",
      "currency":#"EUR"
    }
  },
  "paymentAccountInformation":#{
    "card":#{
      "type":#"003"
    }
  },
  "paymentInformation":#{
    "accountFeatures":#{
      "category":#"AX",
      "group":#"0"
    },
    "tokenizedCard":#{
      "type":#"003"
    },
    "card":#{
      "type":#"003"
    }
  },
  "pointOfSaleInformation":#{
    "emv":#{
      "tags":#"9F2701809F34033F00005F340100910AEE43F0FD6F46AABF3030"
    }
  },
  "processorInformation":#{
    "systemTraceAuditNumber":#"037809",
    "approvalCode":#"437964",

```

```

"networktransactionId":#"000002605437964",
"retrievalReferenceNumber":#"330009037809",
"transactionId":#"000002605437964",
"responseCode":#"00",
"avs":#{
  "code":#"2"
}
},
"reconciliationId":#"6984003567376178404953",
"status":#"AUTHORIZED",
"submitTimeUtc":#"2023-10-27T09:52:39Z"
}

```

Discover Authorization with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#)

Set this field to `TransitDA BAU nominal value auth.`

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyReferenceId](#) Cybersource provides the value for this field.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#) Set this field to `1.00`.

[paymentInformation.card.type](#)

Set this field to `004`.

[pointOfSaleInformation.catLevel](#)

Set this field to `2`.

[pointOfSaleInformation.emv.cardSequenceNumber](#) Set this field to `99`.

[pointOfSaleInformation.emv.tags](#)

[pointOfSaleInformation.entryMode](#)

Set this field to `contactless`.

- pointOfSaleInformation.terminalCapability** Set this field to `5`.
- pointOfSaleInformation.terminalId**
- pointOfSaleInformation.terminalPinCapability** Set this field to `0`.
- pointOfSaleInformation.trackData**
- processingInformation.authorizationOptions.deferredAuthIndicator** Set this field to `true`.
- processingInformation.authorizationOptions.aggregatedAuthIndicator** Set this field to `true`.
- processingInformation.commerceIndicator** Set this field to `retail`.
- processingInformation.industryDataType** Set this field to `transit`.

Related information

- [API Field Reference for the REST API](#)

REST Example: Discover Authorization with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA BAU nominal value auth",
    "code": "123456",
    "transactionId": "1346334405",
    "partner": {
      "solutionId": "123456",
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "capture": "false",
    "commerceIndicator": "retail",
    "authorizationOptions": {
      "deferredAuthIndicator": "true",
      "aggregatedAuthIndicator": "true"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "1.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "004"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",

```



```

    "entryMode": "contactless",
    "terminalCapability": "5",
    "terminalPinCapability": "0",
    "emv": {
      "tags":
"9F2608101F3F75E8596414820211009F360200019F2701409F100A011510000000000000000095050000000000009F370438A87
      "cardSequenceNumber": "99"
    },
    "trackData": ";651000XXXXXXXXXX=491220118040885000000?"
  }
}

```

Response to a Successful Request

```

{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6920241974736435904951/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6920241974736435904951"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6920241974736435904951/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "123456",
    "comments": "TransitDA BAU nominal value auth",
    "partner": {
      "solutionId": "123456"
    }
  },
  "transactionId": "1346334405"
},
"id": "6920241974736435904951",
"orderInformation": {
  "amountDetails": {
    "authorizedAmount": "1.00",
    "currency": "USD"
  }
},
"paymentAccountInformation": {
  "card": {
    "type": "004"
  }
},
"paymentInformation": {
  "accountFeatures": {
    "category": "DI",
    "group": "0"
  }
},
"tokenizedCard": {
  "type": "004"
}

```

```

    },
    "card": {
      "type": "004"
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags": "9F2701409F3501259F36020001"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "033732",
    "approvalCode": "813783",
    "cardReferenceData": "05",
    "networktransactionId": "VISJ 303226529970011",
    "retrievalReferenceNumber": "322614033732",
    "consumerAuthenticationResponse": {
      "code": "0",
      "codeRaw": "0"
    }
  },
  "transactionId": "VISJ 303226529970011",
  "responseCode": "00",
  "avs": {
    "code": "2"
  }
},
"reconciliationId": "6920241974736435904951",
"status": "AUTHORIZED",
"submitTimeUtc": "2023-08-14T14:43:19Z"
}

```

Mastercard Authorization with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#)

Set this field to `TransitDA BAU nominal value auth.`

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

clientReferenceInformation.partner.thirdPartyCybersource	Provides the value for this field.
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
paymentInformation.card.type	Set this field to <code>002</code> .
paymentInformation.initiationChannel	
pointOfSaleInformation.catLevel	Set this field to <code>2</code> .
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set this field to <code>contactless</code> .
pointOfSaleInformation.serviceCode	
pointOfSaleInformation.terminalCapability	Set this field to <code>5</code> .
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to <code>0</code> .
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.addToRetailIndicator	Set this field to <code>true</code> .
processingInformation.authorizationOptions.addToRetailIndicator	Set this field to <code>0</code> .
processingInformation.authorizationOptions.addToRetailIndicator	Set this field to <code>true</code> .
processingInformation.authorizationOptions.transportationMode	
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set this field to <code>retail</code> .
processingInformation.industryDataType	Set this field to <code>transit</code> .

Related information

- [API Field Reference for the REST API](#)

REST Example: Mastercard Authorization with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA BAU nominal value auth",
    "code": "10000568",
    "transactionId": "20000568",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  }
}
```

```

},
"processingInformation": {
  "industryDataType": "transit",
  "commerceIndicator": "retail",
  "capture": "false",
  "captureOptions": {
    "dateToCapture": "0425"
  },
  "authorizationOptions": {
    "authIndicator": "0",
    "deferredAuthIndicator": "true",
    "aggregatedAuthIndicator": "true",
    "transportationMode": "00"
  }
},
"orderInformation": {
  "amountDetails": {
    "totalAmount": "10.00",
    "currency": "EUR"
  }
},
"paymentInformation": {
  "card": {
    "type": "002"
  },
  "initiationChannel": "00"
},
"pointOfSaleInformation": {
  "terminalId": "12345678",
  "catLevel": "2",
  "entryMode": "contactless",
  "terminalCapability": "5",
  "terminalPinCapability": "0",
  "emv": {
    "tags":
"5F2A0209768407A000000000410109F360200039F0306000000000000009C01005F3401019F10120110A0000F040000000000",
  }
},
"trackData": ";5413XXXXXXXXXXXX=49122010123456789?",
"serviceCode": "201"
}
}

```

Response to a Successful Request

```

{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508877845426512004004"
    }
  },
  "clientReferenceInformation": {
    "code": "10000574",
    "partner": {
      "solutionId": "548UHQ8Z"
    }
  },
}

```

```

    "transactionId": "20000574"
  },
  "errorInformation": {
    "reason": "AUTH_DECLINE_CAPTURE_POSSIBLE",
    "message": "Authorization Declined. Follow-on Capture can be processed."
  },
  "id": "6508877845426512004004",
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "164207",
    "networktransactionId": "016153570198200",
    "retrievalReferenceNumber": "211511164207",
    "transactionId": "016153570198200",
    "responseCode": "05",
    "avs": {
      "code": "2"
    }
  },
  "status": "AUTHORIZED"
}

```

Visa Account Verification Request (AVR) with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#)

Set this field to `TransitDA BAU zero value auth.`

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyIdentifier](#) Cybersource provides the value for this field.

clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	Set this field to <code>0.00</code> .
paymentInformation.card.type	Set this field to <code>001</code> .
pointOfSaleInformation.catLevel	Set this field to <code>2</code> .
pointOfSaleInformation.emv.cardSequenceNumber	
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set this field to <code>contactless</code> .
pointOfSaleInformation.terminalCapability	Set this field to <code>5</code> .
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to <code>0</code> .
pointOfSaleInformation.trackData	
processingInformation.commerceIndicator	Set this field to <code>retail</code> .

Related information

- [API Field Reference for the REST API](#)

REST Example: Visa AVR Authorization with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA BAU zero value auth",
    "code": "10000564",
    "transactionId": "20000564",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "capture": "false",
    "commerceIndicator": "retail"
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "0.00",
      "currency": "EUR"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    }
  }
}
```



```

    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "162930",
    "approvalCode": "831000",
    "merchantAdvice": {
      "code": "01",
      "codeRaw": "M001"
    }
  },
  "responseDetails": "ABC",
  "networktransactionId": "016153570198200",
  "retrievalReferenceNumber": "211511162930",
  "consumerAuthenticationResponse": {
    "code": "2",
    "codeRaw": "2"
  },
  "transactionId": "016153570198200",
  "responseCode": "00",
  "avs": {
    "code": "Y",
    "codeRaw": "Y"
  }
},
"reconciliationId": "6508875466126538104002",
"status": "AUTHORIZED",
"submitTimeUtc": "2022-04-25T11:52:26Z"
}

```

Response to a Declined Request

```

{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508876049646556304003"
    }
  },
  "clientReferenceInformation": {
    "code": "10000566",
    "partner": {
      "solutionId": "548UHQ8Z"
    }
  },
  "transactionId": "20000566"
},
"errorInformation": {
  "reason": "AUTH_DECLINE_CAPTURE_POSSIBLE",
  "message": "Authorization Declined. Follow-on Capture can be processed."
},
"id": "6508876049646556304003",
"pointOfSaleInformation": {
  "emv": {
    "tags":
"9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
  }
}
}

```



```

},
"processorInformation": {
  "systemTraceAuditNumber": "162936",
  "networktransactionId": "016153570198200",
  "retrievalReferenceNumber": "211511162936",
  "transactionId": "016153570198200",
  "responseCode": "05",
  "avs": {
    "code": "2"
  }
},
"status": "AUTHORIZED"
}
    
```

Required Fields for Sales with EMV Data

This table provides information about the fields required for sales with EMV data.

Sales with EMV Data Fields

REST API Field	Discover Sale	Visa Deferred	Information/Value
clientReferenceInformation.code			—
clientReferenceInformation.comments			For this value, see Transaction Types on page 48.
clientReferenceInformation.partner.solutionId			Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber			Cybersource provides the value for this field.
clientReferenceInformation.transactionId			—
orderInformation.amountDetails.currency			—
orderInformation.amountDetails.totalAmount			For Visa AVR, set this field to 0.00.
paymentInformation.card.type			—
pointOfSaleInformation.catLevel			Set this field to 2.
pointOfSaleInformation.emv.cardSequenceNumber			—

REST API Field	Discover Sale	Visa Defered	Information/Value
pointOfSaleInformation.emv.tags			—
pointOfSaleInformation.entryMode			Set this field to contactless.
pointOfSaleInformation.terminalCapability			Set this field to 5.
pointOfSaleInformation.terminalId			—
pointOfSaleInformation.terminalPinCapability			Set this field to 0.
pointOfSaleInformation.trackData			—
processingInformation.authorizationOptions.aggregatedAuthIndicator			Set this field to true.
processingInformation.authorizationOptions.deferredAuthIndicator			Set this field to true.
processingInformation.capture			Set this field to true.
processingInformation.captureOptions.dateToCapture			—
processingInformation.commerceIndicator			Set this field to retail.
processingInformation.industryDataType			Set this field to transit.

Discover Sale with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

clientReferenceInformation.comments	Set this field to <code>TransitDA BAU nominal value sale</code> .
clientReferenceInformation.partner.solutionId	Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyId	Cybersource provides the value for this field.
clientReferenceInformation.transactionId	
orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	Set this field to <code>1.00</code> .
paymentInformation.card.type	Set this field to <code>004</code> .
pointOfSaleInformation.catLevel	Set this field to <code>2</code> .
pointOfSaleInformation.emv.cardSequenceNumber	Set this field to <code>99</code> .
pointOfSaleInformation.emv.tags	
pointOfSaleInformation.entryMode	Set this field to <code>contactless</code> .
pointOfSaleInformation.terminalCapability	Set this field to <code>5</code> .
pointOfSaleInformation.terminalId	
pointOfSaleInformation.terminalPinCapability	Set this field to <code>0</code> .
pointOfSaleInformation.trackData	
processingInformation.authorizationOptions.addToAccount	Set this field to <code>true</code> .
processingInformation.authorizationOptions.addToCard	Set this field to <code>true</code> .
processingInformation.capture	Set the value to <code>true</code> .
processingInformation.captureOptions.dateToCapture	
processingInformation.commerceIndicator	Set this field to <code>retail</code> .
processingInformation.industryDataType	Set this field to <code>transit</code> .

Related information

- [API Field Reference for the REST API](#)

REST Example: Discover Sale with EMV Data

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA BAU full value sale",
    "code": "123456",
    "transactionId": "1357334401",
    "partner": {
      "solutionId": "123456",
```

```

    "thirdPartyCertificationNumber": "123456789012"
  }
},
"processingInformation": {
  "industryDataType": "transit",
  "reconciliationId": "123456789",
  "captureOptions": {
    "dateToCapture": "0818"
  },
  "capture": "true",
  "commerceIndicator": "retail"
},
"orderInformation": {
  "amountDetails": {
    "totalAmount": "25.00",
    "currency": "USD"
  }
},
"paymentInformation": {
  "card": {
    "type": "004"
  }
},
"pointOfSaleInformation": {
  "terminalId": "12345678",
  "catLevel": "2",
  "entryMode": "contactless",
  "terminalCapability": "5",
  "terminalPinCapability": "0",
  "emv": {
    "tags":
"9F2608101F3F75E8596414820211009F360200019F2701409F100A01151000000000000000095050000000000009F370438A87
    "cardSequenceNumber": "99"
  },
  "trackData": ";651000XXXXXXXXXX=491220118040885000000?"
}
}
}

```

Response to a Successful Request

```

{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/payments/6920243246666458104951/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6920243246666458104951"
    }
  },
  "clientReferenceInformation": {
    "code": "123456",
    "comments": "TransitDA BAU full value sale",
    "partner": {
      "solutionId": "123456"
    }
  }
}

```

```

    },
    "transactionId": "1357334401"
  },
  "id": "6920243246666458104951",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "15.00",
      "authorizedAmount": "15.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "004"
    }
  },
  "paymentInformation": {
    "accountFeatures": {
      "category": "DI",
      "group": "0"
    },
    "tokenizedCard": {
      "type": "004"
    },
    "card": {
      "type": "004"
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags": "9F2701409F3501259F36020001"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "033735",
    "approvalCode": "378857",
    "cardReferenceData": "05",
    "networktransactionId": "VISJ 303226531251404",
    "retrievalReferenceNumber": "322614033735",
    "consumerAuthenticationResponse": {
      "code": "0",
      "codeRaw": "0"
    },
    "transactionId": "VISJ 303226531251404",
    "responseCode": "00",
    "avs": {
      "code": "2"
    }
  },
  "reconciliationId": "6920243246666458104951",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2023-08-14T14:45:26Z"
}

```

Visa Deferred Sale with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#) Set this field to `TransitDA BAU full value sale`.

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyIdentifier](#) Cybersource provides the value for this field.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.type](#) Set this field to `001`.

[pointOfSaleInformation.catLevel](#) Set this field to `2`.

[pointOfSaleInformation.emv.cardSequenceNumber](#)

[pointOfSaleInformation.emv.tags](#)

[pointOfSaleInformation.entryMode](#) Set this field to `contactless`.

[pointOfSaleInformation.terminalCapability](#) Set this field to `5`.

[pointOfSaleInformation.terminalId](#)

[pointOfSaleInformation.terminalPinCapability](#) Set this field to `0`.

[pointOfSaleInformation.trackData](#)

[processingInformation.authorizationOptions.authorizationIndicator](#) Set this field to `true`.

[processingInformation.authorizationOptions.authorizationMode](#) Set this field to `true`.

[processingInformation.capture](#) Set this field to `true`.

[processingInformation.captureOptions.dateToCapture](#)


```

    },
    "trackData": ";4761XXXXXXXXXXXX=241220119058254?"
  }
}

```

Response to a Successful Request

```

{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/payments/6508875814676551204001/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508875814676551204001"
    }
  },
  "clientReferenceInformation": {
    "code": "10000565",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000565"
  },
  "id": "6508875814676551204001",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "authorizedAmount": "10.00",
      "currency": "EUR"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "accountFeatures": {
      "group": "0"
    },
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
    }
  },
  "processorInformation": {

```



```

"systemTraceAuditNumber": "164186",
"approvalCode": "831000",
"networktransactionId": "016153570198200",
"retrievalReferenceNumber": "211511164186",
"transactionId": "016153570198200",
"responseCode": "00",
"avs": {
  "code": "2"
}
},
"reconciliationId": "6508875814676551204001",
"status": "AUTHORIZED",
"submitTimeUtc": "2022-04-25T11:53:01Z"
}

```

Response to a Declined Request with First Ride Protection

```

{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508878333386555704002"
    }
  },
  "clientReferenceInformation": {
    "code": "10000576",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000576"
  },
  "errorInformation": {
    "reason": "AUTH_DECLINE_CAPTURE_POSSIBLE",
    "message": "Authorization Declined. Follow-on Capture can be processed."
  },
  "id": "6508878333386555704002",
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "164212",
    "networktransactionId": "016153570198200",
    "retrievalReferenceNumber": "211511164212",
    "transactionId": "016153570198200",
    "responseCode": "05",
    "avs": {
      "code": "2"
    }
  },
  "status": "DECLINED"
}

```

Required Fields for an Authorization for Debt Recovery

This table provides information about the fields required for tap-initiated and merchant-initiated (MIT) authorizations for debt recovery.

Authorization for Debt Recovery Fields

REST API Field	Tap	MIT	Information/Value
clientReferenceInformation.code			—
clientReferenceInformation.comments			For this value, see Transaction Types on page 48.
clientReferenceInformation.partner.solutionId			Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber			Cybersource provides the value for this field.
clientReferenceInformation.transactionId			—
orderInformation.amountDetails.currency			—
orderInformation.amountDetails.totalAmount			—
paymentInformation.card.expirationMonth	—		—
paymentInformation.card.expirationYear	—		—
paymentInformation.card.number	—		—
paymentInformation.card.type			—
pointOfSaleInformation.catLevel		—	Set this field to 2.
pointOfSaleInformation.emv.cardSequenceNumber		—	Set this field to 1.
pointOfSaleInformation.emv.tags		—	—

REST API Field	Tap	MIT	Information/Value
pointOfSaleInformation.entryMode		—	Set this field to contact less.
pointOfSaleInformation.terminalCapability		—	Set this field to 5.
pointOfSaleInformation.terminalId		—	—
pointOfSaleInformation.terminalPinCapability		—	Set this field to 0.
pointOfSaleInformation.trackData		—	—
processingInformation.authorizationOptions.debtRecoveryIndicator			Set this field to true.
processingInformation.authorizationOptions.deferredAuthIndicator		—	Set this field to true.
processingInformation.authorizationOptions.ignoreAvsResult	—		Set this field to true.
processingInformation.authorizationOptions.ignoreCvResult	—		Set this field to true.
processingInformation.authorizationOptions.initiator.credentialStoredOnFile	—		Set this field to false.
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionId	—		—
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	—		Set this field to 1.
processingInformation.authorizationOptions.initiator.storedCredentialUsed	—		Set this field to true.
processingInformation.authorizationOptions.initiator.type	—		Set this field to merchant.
processingInformation.captureOptions.dateToCapture		—	—

REST API Field	Tap	MIT	Information/Value
processingInformation.commerceIndicator			Set this field to retail.
processingInformation.industryDataType			Set this field to transit.

Tap-Initiated Authorization for Debt Recovery with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#)

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

[clientReferenceInformation.partner.solutionId](#)

Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyId](#)

Cybersource provides the value for this field.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.type](#)

[pointOfSaleInformation.catLevel](#)

Set this field to `2`.

[pointOfSaleInformation.emv.cardSequenceNumber](#)

Set this field to `1`.

[pointOfSaleInformation.emv.tags](#)

[pointOfSaleInformation.entryMode](#)

Set this field to `contactless`.

<code>pointOfSaleInformation.terminalCapability</code>	Set this field to 5.
<code>pointOfSaleInformation.terminalId</code>	
<code>pointOfSaleInformation.terminalPinCapability</code>	Set this field to 0.
<code>pointOfSaleInformation.trackData</code>	
<code>processingInformation.authorizationOptions</code>	Set this field to true.
<code>processingInformation.authorizationOptions</code>	Set this field to true.
<code>processingInformation.captureOptions.dateToCapture</code>	
<code>processingInformation.commerceIndicator</code>	Set this field to retail.
<code>processingInformation.industryDataType</code>	Set this field to transit.

Related information

- [API Field Reference for the REST API](#)

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

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA Debt recovery tap auth",
    "code": "10000597",
    "transactionId": "20000597",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "captureOptions": {
      "dateToCapture": "0425"
    },
    "commerceIndicator": "retail",
    "authorizationOptions": {
      "debtRecoveryIndicator": "true",
      "deferredAuthIndicator": "true"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    }
  }
}
```


Merchant-Initiated Authorizations for Debt Recovery with Stored Card Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#) Set this field to `TransitDA Debt recovery MIT auth.`

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyCertificationId](#) Cybersource provides the value for this field.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.expirationMonth](#)

[paymentInformation.card.expirationYear](#)

[paymentInformation.card.number](#)

[paymentInformation.card.type](#)

[processingInformation.authorizationOptions.authorizationResult](#) Set this field to `true`.

[processingInformation.authorizationOptions.authorizationResult](#) Set this field to `true`.

[processingInformation.authorizationOptions.authorizationResult](#) Set this field to `true`.

[processingInformation.authorizationOptions.authorizationResult](#) Set this field to `false`. [merchantInitiatedTransaction.previousTransactionId](#)

[processingInformation.authorizationOptions.authorizationResult](#) Set this field to `true`. [merchantInitiatedTransaction.reason](#)

[processingInformation.authorizationOptions.authorizationResult](#) Set this field to `1`. [merchantInitiatedTransaction.reason](#)

[processingInformation.authorizationOptions.authorizationResult](#) Set this field to `true`. [merchantInitiatedTransaction.reason](#)

processingInformation.authorizationOptions Set this field to `merchant`.

processingInformation.commerceIndicator Set this field to `moto`.

processingInformation.industryDataType Set this field to `transit`.

Related information

- [API Field Reference for the REST API](#)

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

Request

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



```

}
}
}

```

Response to a Declined Request

```

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

```

Required Fields for a Sale for Debt Recovery

This table provides information about the fields required for tap-initiated and merchant-initiated (MIT) sales for debt recovery.

Sales for Debt Recovery Fields

REST API Field	Tap	MIT	Information/Value
clientReferenceInformation.code			—
clientReferenceInformation.comments			For this value, see Transaction Types on page 48.
clientReferenceInformation.partner.solutionId			Cybersource provides the value for this field.
clientReferenceInformation.partner.thirdPartyCertificationNumber			Cybersource provides the value for this field.
clientReferenceInformation.transactionId			—
orderInformation.amountDetails.currency			—
orderInformation.amountDetails.totalAmount			—
paymentInformation.card.expirationMonth	—		—
paymentInformation.card.expirationYear	—		—
paymentInformation.card.number	—		—
paymentInformation.card.type			—
paymentInformation.initiationChannel		—	Set this field to 00.
pointOfSaleInformation.catLevel		—	Set this field to 2.
pointOfSaleInformation.emv.tags		—	—
pointOfSaleInformation.entryMode		—	Set this field to contactless.
pointOfSaleInformation.serviceCode		—	—
pointOfSaleInformation.terminalCapability		—	Set this field to 5.

REST API Field	Tap	MIT	Information/Value
pointOfSaleInformation.terminalId		—	—
pointOfSaleInformation.terminalPinCapability		—	Set this field to 0.
pointOfSaleInformation.trackData		—	—
processingInformation.authorizationOptions.authIndicator			Set this field to 1.
processingInformation.authorizationOptions.debtRecoveryIndicator			Set this field to true.
processingInformation.authorizationOptions.deferredAuthIndicator		—	Set this field to true. Do not include for Mastercard transactions.
processingInformation.authorizationOptions.ignoreAvsResult	—		Set this field to true.
processingInformation.authorizationOptions.ignoreCvResult	—		Set this field to true.
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionId	—		—
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	—		Set this field to 1.
processingInformation.authorizationOptions.initiator.storedCredentialUsed	—		Set this field to true.
processingInformation.authorizationOptions.initiator.type	—		Set this field to merchant.
processingInformation.authorizationOptions.transportationMode			—
processingInformation.capture			Set this field to true.

REST API Field	Tap	MIT	Information/Value
processingInformation.captureOptions.dateToCapture		—	—
processingInformation.commerceIndicator			For tap-initiated, set this field to retail. For merchant-initiated, set this field to moto.
processingInformation.industryDataType			Set this field to transit.

Tap-Initiated Sale for Mastercard Debt Recovery with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#)

Set this field to `TransitDA Debt recovery tap sale`.

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyIdentifier](#) Cybersource provides the value for this field.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.type](#)

[paymentInformation.initiationChannel](#)

Set this field to `00`.

<code>pointOfSaleInformation.catLevel</code>	Set this field to <code>2</code> .
<code>pointOfSaleInformation.emv.tags</code>	
<code>pointOfSaleInformation.entryMode</code>	Set this field to <code>contactless</code> .
<code>pointOfSaleInformation.serviceCode</code>	
<code>pointOfSaleInformation.terminalCapability</code>	Set this field to <code>5</code> .
<code>pointOfSaleInformation.terminalId</code>	
<code>pointOfSaleInformation.terminalPinCapability</code>	Set this field to <code>0</code> .
<code>pointOfSaleInformation.trackData</code>	
<code>processingInformation.authorizationOptions</code>	Set this field to <code>1</code> .
<code>processingInformation.authorizationOptions</code>	Set this field to <code>true</code> .
<code>processingInformation.authorizationOptions.transportationMode</code>	
<code>processingInformation.capture</code>	Set this field to <code>true</code> .
<code>processingInformation.captureOptions.dateToCapture</code>	
<code>processingInformation.commerceIndicator</code>	Set this field to <code>retail</code> .
<code>processingInformation.industryDataType</code>	Set this field to <code>transit</code> .

Related information

- [API Field Reference for the REST API](#)

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

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA Debt recovery tap sale",
    "code": "10000575MC",
    "transactionId": "20000575MC",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "commerceIndicator": "retail",
    "capture": "true",
    "captureOptions": {
      "dateToCapture": "0425"
    }
  },
  "authorizationOptions": {
    "authIndicator": "1",

```



```

}
},
"processorInformation": {
  "systemTraceAuditNumber": "162956",
  "networktransactionId": "016153570198200",
  "retrievalReferenceNumber": "211511162956",
  "transactionId": "016153570198200",
  "responseCode": "05",
  "avs": {
    "code": "2"
  }
},
"status": "DECLINED"
}

```

Tap-Initiated Sale for Visa Debt Recovery with EMV Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#)

Set this field to `TransitDA Debt recovery tap sale`.

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyId](#) Cybersource provides the value for this field.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.type](#)

[paymentInformation.initiationChannel](#)

Set this field to `00`.

<code>pointOfSaleInformation.catLevel</code>	Set this field to <code>2</code> .
<code>pointOfSaleInformation.emv.tags</code>	
<code>pointOfSaleInformation.entryMode</code>	Set this field to <code>contactless</code> .
<code>pointOfSaleInformation.serviceCode</code>	
<code>pointOfSaleInformation.terminalCapability</code>	Set this field to <code>5</code> .
<code>pointOfSaleInformation.terminalId</code>	
<code>pointOfSaleInformation.terminalPinCapability</code>	Set this field to <code>0</code> .
<code>pointOfSaleInformation.trackData</code>	
<code>processingInformation.authorizationOptions</code>	Set this field to <code>1</code> .
<code>processingInformation.authorizationOptions</code>	Set this field to <code>true</code> .
<code>processingInformation.authorizationOptions</code>	Set this field to <code>true</code> .
<code>processingInformation.authorizationOptions.transportationMode</code>	
<code>processingInformation.capture</code>	Set this field to <code>true</code> .
<code>processingInformation.captureOptions.dateToCapture</code>	
<code>processingInformation.commerceIndicator</code>	Set this field to <code>retail</code> .
<code>processingInformation.industryDataType</code>	Set this field to <code>transit</code> .

Related information

- [API Field Reference for the REST API](#)

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

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA Debt recovery tap sale",
    "code": "10000575",
    "transactionId": "20000575",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "commerceIndicator": "retail",
    "capture": "true",
    "captureOptions": {
      "dateToCapture": "0425"
    }
  }
}
```



```

    "authorizationOptions": {
      "authIndicator": "1",
      "debtRecoveryIndicator": "true",
      "deferredAuthIndicator": "true",
      "transportationMode": "00"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    },
    "initiationChannel": "00"
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",
    "entryMode": "contactless",
    "terminalCapability": "5",
    "terminalPinCapability": "0",
    "emv": {
      "tags":
"5F2A0209768407A00000000410109F360200039F030600000000000009C01005F3401019F10120110A0000F040000000000
    },
    "trackData": ";4413XXXXXXXXXXXX=49122010123456789?",
    "serviceCode": "201"
  }
}

```

Response to a Declined Request

```

{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508878027476513004004"
    }
  },
  "clientReferenceInformation": {
    "code": "10000575",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000575"
  },
  "errorInformation": {
    "reason": "PROCESSOR_DECLINED",
    "message": "Decline - General decline of the card. No other information provided by the issuing bank."
  },
  "id": "6508878027476513004004",
  "pointOfSaleInformation": {

```

```

    "emv": {
      "tags":
"9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "162956",
    "networktransactionId": "016153570198200",
    "retrievalReferenceNumber": "211511162956",
    "transactionId": "016153570198200",
    "responseCode": "05",
    "avs": {
      "code": "2"
    }
  },
  "status": "DECLINED"
}

```

Merchant-Initiated Sale for Debt Recovery with Stored Card Data

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

Endpoint

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

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

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

[clientReferenceInformation.code](#)

[clientReferenceInformation.comments](#)

Set this field to `TransitDA Debt recovery MIT sale`.

[clientReferenceInformation.partner.solutionId](#) Cybersource provides the value for this field.

[clientReferenceInformation.partner.thirdPartyId](#) Cybersource provides the value for this field.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.expirationMonth](#)

`paymentInformation.card.expirationYear`

`paymentInformation.card.number`

`paymentInformation.card.type`

`processingInformation.authorizationOptions`. Set this field to 1.

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

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

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

`processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction`.

`previousTransactionId`

`processingInformation.authorizationOptions`. Set this field to 1.

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

`processingInformation.authorizationOptions`. Set this field to merchant.

`processingInformation.authorizationOptions.transportationMode`

`processingInformation.capture` Set this field to true.

`processingInformation.commerceIndicator` Set this field to moto.

`processingInformation.industryDataType` Set this field to transit.

Related information

- [API Field Reference for the REST API](#)

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

Request

```
{
  "clientReferenceInformation": {
    "comments": "TransitDA Debt recovery MIT sale",
    "code": "10000579",
    "transactionId": "20000579",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "commerceIndicator": "moto",
    "industryDataType": "transit",
    "reconciliationId": "1111",
    "capture": "true",
    "authorizationOptions": {
      "debtRecoveryIndicator": "true",
      "authIndicator": "1",

```

```

    "ignoreAvsResult": "true",
    "ignoreCvResult": "true",
    "transportationMode": "01",
    "initiator": {
      "type": "merchant",
      "storedCredentialUsed": "true",
      "merchantInitiatedTransaction": {
        "reason": "1",
        "previousTransactionId": "016153570198200"
      }
    }
  },
  "paymentInformation": {
    "card": {
      "number": "541333XXXXXXXXXX",
      "expirationMonth": "12",
      "expirationYear": "2049",
      "type": "002"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "EUR"
    }
  }
}

```

Response to a Declined Request

```

{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6508879024886569004002"
    }
  },
  "clientReferenceInformation": {
    "code": "10000579",
    "partner": {
      "solutionId": "548UHQ8Z"
    },
    "transactionId": "20000579"
  },
  "errorInformation": {
    "reason": "PROCESSOR_DECLINED",
    "message": "Decline - General decline of the card. No other information provided by the issuing bank."
  },
  "id": "6508879024886569004002",
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"9F36020015910AB58D60185BEF0247303072179F180430303031860E04DA9F580903B1BAEDFD1438BA48"
    }
  },
}

```

```
"processorInformation": {  
  "systemTraceAuditNumber": "162971",  
  "networktransactionId": "016153570198200",  
  "retrievalReferenceNumber": "211511162971",  
  "transactionId": "016153570198200",  
  "responseCode": "05",  
  "avs": {  
    "code": "1"  
  }  
},  
"status": "DECLINED"  
}
```

Mass Transit Payment Services Using TMS Tokens

You can use TMS tokens to request these mass transit payment services:

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

In the card-present EMV contactless request, include the transient token ID in the **tokenInformation.jti** field in place of track 2 data.

When making a tap token creation request, you can include EMV tag-length-value (TLV) tags in the **paymentInformation.fluidData.value** field or as part of the payment transaction request within the **pointOfSaleInformation.emv.tags** field.

If you send EMV tags in the tap token create request, do not send EMV tags in the payment transaction request.

When EMV TLV tags are present in both the payment transaction and the token vault, Cybersource reads the value provided in the payment transaction rather than the values stored in the vault.

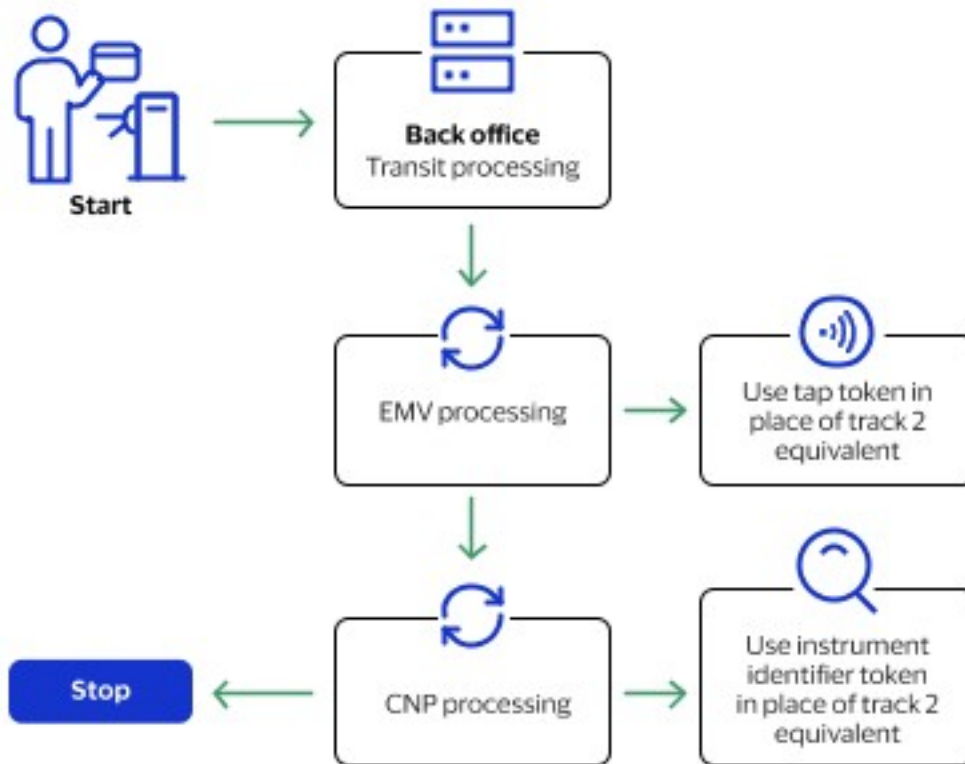
Mastercard EMV transactions include three field values that can be handled automatically:

- **paymentInformation.card.initiationChannel**
- **pointOfSaleInformation.emv.cardSequenceNumber**
- **pointOfSaleInformation.serviceCode**

Your account can be configured to read these values automatically from the EMV TLV tags and track 2 equivalent. When that option is enabled, do not include those three fields in EMV payment requests.

If any of these values are present in both the separate fields and the EMV TLV and track 2 equivalent, Cybersource reads the value provided in the separate fields rather than the values present in the EMV TLV and track 2 equivalent.

Merchant initiated card-not-present payment requests use the **paymentInformation.instrumentIdentifier.id** field instead of the card number.



Payment Processing with a Token

Mass Transit Mastercard Authorization with a Token

This section describes how to process an authorization with a token.

Endpoint

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

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

Required Fields for a Mastercard Authorization with a Token Using the REST API

[clientReferenceInformation.code](#)

[clientReferenceInformation.partner.solutionId](#) The value for this field is provided by Cybersource.

[clientReferenceInformation.partner.thirdPartyId](#) The value for this field is provided by Cybersource.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

orderInformation.amountDetails.totalAmount

paymentInformation.initiationChannel

pointOfSaleInformation.catLevel Set this field to 2.

pointOfSaleInformation.entryMode Set this field to `contactless`.

pointOfSaleInformation.terminalCapability Set this field to 5.

pointOfSaleInformation.terminalId

pointOfSaleInformation.terminalPinCapability Set this field to 0.

processingInformation.authorizationOptions.authorizationIndicator Set this field to `true`.

processingInformation.authorizationOptions.authorizationMode Set this field to 0.

processingInformation.authorizationOptions.authorizationType Set this field to `retail` or `transit`.

processingInformation.authorizationOptions.transportationMode

processingInformation.captureOptions.dateToCapture

processingInformation.commerceIndicator Set this field to `retail`.

processingInformation.industryDataType Set this field to `transit`.

tokenInformation.jti Generate this value using the UUID2/UUID4 data format.

Related Information

- [API field reference guide for the REST API](#)

Related information

- [API Field Reference for the REST API](#)

REST Example: Performing a Mastercard Authorization with a Token

Request

```
{
  "clientReferenceInformation": {
    "code": "10000721",
    "transactionId": "987654321",
    "partner": {
      "thirdPartyCertificationNumber": "BPCDRC220403",
      "solutionId": "548UHQ8Z"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "processingInformation.commerceIndicator": "retail",
    "captureOptions": {
      "dateToCapture": "0425"
    }
  }
}
```



```

    "authorizationOptions": {
      "authIndicator": "0",
      "deferredAuthIndicator": "true",
      "aggregatedAuthIndicator": "true",
      "transportationMode": "00"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "initiationChannel": "00"
  },
  "tokenInformation": {
    "jti": "a76392f4-cde4-97aa-1112-0242ac14c005"
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",
    "entryMode": "contactless",
    "terminalCapability": "5",
    "terminalPinCapability": "0"
  }
}

```

Response to a Successful Request

```

{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/64823013154065933040011/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6482301315406593304011"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6482301315406593304011/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "10000721",
    "partner": {
      "solutionId": "548UHQ8Z"
    }
  },
  "transactionId": "987654321"
},
  "id": "6482301315406593304011",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "10.00",

```

```

    "currency": "THB"
  }
},
"paymentAccountInformation": {
  "card": {
    "type": "002"
  }
},
"paymentInformation": {
  "tokenizedCard": {
    "type": "002"
  },
  "card": {
    "type": "002"
  }
},
"processorInformation": {
  "systemTraceAuditNumber": "191316",
  "approvalCode": "831000",
  "merchantAdvice": {
    "code": "01",
    "codeRaw": "M001"
  },
  "responseDetails": "ABC",
  "networktransactionId": "016153570198201",
  "retrievalReferenceNumber": "211511164721",
  "consumerAuthenticationResponse": {
    "code": "2",
    "codeRaw": "2"
  },
  "transactionId": "016153570198201",
  "responseCode": "00",
  "avs": {
    "code": "Y",
    "codeRaw": "Y"
  }
},
"reconciliationId": "6482301315406593304011",
"status": "AUTHORIZED",
"submitTimeUtc": "2022-03-25T17:42:11Z"
}

```

Mass Transit Visa Account Verification Request (AVR) with a Token

This section describes how to process a zero amount authorization for a mass transit Visa account verification request (AVR).

Endpoint

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

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

Required Fields for a Visa AVR Authorization with a Transient Token Using the REST API

<code>clientReferenceInformation.partner.thirdPartyCertificationNumber</code>	
<code>clientReferenceInformation.transactionId</code>	
<code>orderInformation.amountDetails.currency</code>	
<code>orderInformation.amountDetails.totalAmount</code>	Set this field to <code>0.00</code> .
<code>paymentInformation.card.type</code>	Set this field to <code>001</code> .
<code>pointOfSaleInformation.catLevel</code>	Set this field to <code>2</code> .
<code>pointOfSaleInformation.entryMode</code>	Set this field to <code>contactless</code> .
<code>pointOfSaleInformation.terminalCapability</code>	Set this field to <code>5</code> .
<code>pointOfSaleInformation.terminalId</code>	
<code>pointOfSaleInformation.terminalPinCapability</code>	Set this field to <code>0</code> .
<code>processingInformation.commerceIndicator</code>	Set this field to <code>retail</code> .
<code>processingInformation.reconciliationId</code>	
<code>tokenInformation.jti</code>	

Related information

- [API Field Reference for the REST API](#)

REST Example: Performing a Visa AVR Authorization with a Transient Token

Request

```
{
  "clientReferenceInformation": {
    "transactionId": "5987654321",
    "partner": {
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "processingInformation": {
    "processingInformation.commerceIndicator": "retail",
    "reconciliationId": "asfafafaffa"
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "0.00",
      "currency": "THB"
    }
  }
},
```

```

"tokenInformation": {
  "jti": "a76392f4-cde4-97aa-1112-0242ac14c005"
},
"pointOfSaleInformation": {
  "terminalId": "12345678",
  "catLevel": "2",
  "entryMode": "contactless",
  "terminalCapability": "5",
  "terminalPinCapability": "0"
}
}

```

Response to a Successful Request

```

{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6482301315406593304003/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6482301315406593304003"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6482301315406593304003/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "123456",
    "transactionId": "5987654321"
  },
  "id": "6482301315406593304003",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "0.00",
      "currency": "THB"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "191316",
    "approvalCode": "831000",

```

```

"merchantAdvice": {
  "code": "01",
  "codeRaw": "M001"
},
"responseDetails": "ABC",
"networktransactionId": "016153570198200",
"consumerAuthenticationResponse": {
  "code": "2",
  "codeRaw": "2"
},
"transactionId": "016153570198200",
"responseCode": "00",
"avs": {
  "code": "Y",
  "codeRaw": "Y"
}
},
"reconciliationId": "asfafafaffa",
"status": "AUTHORIZED",
"submitTimeUtc": "2022-03-25T17:42:11Z"
}

```

Mass Transit Visa Deferred Sale with a Token

This section describes how to process a Visa deferred sale.

A sale transaction is a bundled authorization and capture. At the end of the travel period, request a Visa deferred sale with a token for an aggregated payment.

Endpoint

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

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

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

[clientReferenceInformation.partner.thirdPartyCertificationNumber](#)

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[pointOfSaleInformation.catLevel](#) Set this field to **2**.

[pointOfSaleInformation.entryMode](#) Set this field to **contactless**.

[pointOfSaleInformation.terminalCapability](#) Set this field to **5**.

[pointOfSaleInformation.terminalId](#)

[pointOfSaleInformation.terminalPinCapability](#) Set this field to **0**.

processingInformation.authorizationOptions.deferredAuthIndicator Set this field to `true`.

processingInformation.authorizationOptions.aggregatedAuthIndicator Set this field to `true`.

processingInformation.capture Set this field to `true`.

processingInformation.captureOptions.dateToCapture

processingInformation.commerceIndicator Set this field to `retail`.

processingInformation.industryDataType Set this field to `transit`.

processingInformation.reconciliationId

tokenInformation.jti

Related information

- API Field Reference for the REST API

REST Example: Performing a Visa Deferred Sale with a Token

Request

```
{
  "clientReferenceInformation": {
    "transactionId": "21987654321",
    "partner": {
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "reconciliationId": "fgssgsgsgsfg",
    "captureOptions": {
      "dateToCapture": "0325"
    },
    "capture": "true",
    "processingInformation.commerceIndicator": "retail",
    "authorizationOptions": {
      "deferredAuthIndicator": "true",
      "aggregatedAuthIndicator": "true"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "300.00",
      "currency": "THB"
    }
  },
  "tokenInformation": {
    "jti": "a76392f4-cde4-97aa-1111-0242ac14c005"
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",
    "entryMode": "contactless",
  }
}
```

```

    "terminalCapability": "5",
    "terminalPinCapability": "0"
  }
}

```

Response to a Successful Request

```

{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/payments/6482303624466600104004/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6482303624466600104004"
    }
  },
  "clientReferenceInformation": {
    "code": "testcode1012",
    "transactionId": "21987654321"
  },
  "id": "6482303624466600104004",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "300.00",
      "authorizedAmount": "300.00",
      "currency": "THB"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "processorInformation": {
    "systemTraceAuditNumber": "191328",
    "approvalCode": "831000",
    "merchantAdvice": {
      "code": "01",
      "codeRaw": "M001"
    }
  },
  "responseDetails": "ABC",
  "networktransactionId": "016153570198200",
  "consumerAuthenticationResponse": {
    "code": "2",
    "codeRaw": "2"
  },
}

```

```

"transactionId": "016153570198200",
"responseCode": "00",
"avs": {
  "code": "Y",
  "codeRaw": "Y"
}
},
"reconciliationId": "fgssgsgsgsf",
"status": "AUTHORIZED",
"submitTimeUtc": "2022-03-25T17:46:02Z"
}

```

Mass Transit Authorizations for a Tap-Initiated Debt Recovery with a Token

This section describes how to process a tap-initiated authorization for debt recovery with a token.

When a cardholder attempts to use a blocked card at the transit reader, create a fresh debt recovery authorization request using the chip data from the new tap, along with the fare amount of the previous declined authorization.

Endpoint

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

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

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

[clientReferenceInformation.partner.thirdPartyCertificationNumber](#)

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.type](#)

[pointOfSaleInformation.catLevel](#) Set this field to 2.

[pointOfSaleInformation.entryMode](#) Set this field to `contactless`.

[pointOfSaleInformation.terminalCapability](#) Set this field to 5.

[pointOfSaleInformation.terminalId](#)

[pointOfSaleInformation.terminalPinCapability](#) Set this field to 0.

[processingInformation.authorizationOptions](#) Set this field to `true`.

[processingInformation.authorizationOptions](#) Set this field to `true`.

[processingInformation.captureOptions.dateToCapture](#)

[processingInformation.commerceIndicator](#) Set this field to `retail`.

[processingInformation.industryDataType](#) Set this field to `transit`.

[processingInformation.reconciliationId](#)

[tokenInformation.jti](#)

Related information

- [API Field Reference for the REST API](#)

REST Example: Performing a Tap-Initiated Authorization for Debt Recovery with a Token

Request

```
{
  "clientReferenceInformation": {
    "transactionId": "9987654321",
    "partner": {
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "reconciliationId": "dsgfsgsgsfdsgf",
    "captureOptions": {
      "dateToCapture": "0114"
    },
    "processingInformation.commerceIndicator": "retail",
    "authorizationOptions": {
      "debtRecoveryIndicator": "true",
      "deferredAuthIndicator": "true"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "THB"
    }
  },
  "paymentInformation": {
    "card": {
      "type": "001"
    }
  },
  "tokenInformation": {
    "jti": "a76392f4-cde4-97aa-1111-0242ac14c005"
  },
  "pointOfSaleInformation": {
    "terminalId": "12345678",
    "catLevel": "2",
    "entryMode": "contactless",
  }
}
```

```

    "terminalCapability": "5",
    "terminalPinCapability": "0"
  }
}

```

Response to a Successful Request

```

{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6502823788756237404002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6502823788756237404002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6502823788756237404002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "123456",
    "transactionId": "9987654321"
  },
  "id": "6502823788756237404002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "10.00",
      "currency": "THB"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "accountFeatures": {
      "category": "A",
      "group": "0"
    },
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags": "5004564953419F26087C14E9BE1F1065094F07A0000000031010820220009F360203709F0702C0809F2701409"
    }
  },
  "processorInformation": {

```

```

"systemTraceAuditNumber": "115052",
"approvalCode": "831000",
"networktransactionId": "016153570198200",
"retrievalReferenceNumber": "210811115052",
"transactionId": "016153570198200",
"responseCode": "00",
"avs": {
  "code": "2"
}
},
"reconciliationId": "dsgfsgsgsfdsgf",
"status": "AUTHORIZED",
"submitTimeUtc": "2022-04-18T11:46:19Z"
}

```

Mass Transit Merchant-Initiated Authorizations for Debt Recovery with a Token

This section describes how to process an authorization for a merchant-initiated debt recovery with a token.

Endpoint

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

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

Required Fields for a Merchant-Initiated Authorization for Debt Recovery with a Token Using the REST API

[clientReferenceInformation.partner.thirdPartyCertificationNumber](#)

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.expirationMonth](#)

[paymentInformation.card.expirationYear](#)

[paymentInformation.card.type](#)

[paymentInformation.instrumentIdentifier.id](#)

[processingInformation.authorizationOptions.ignoreAvsResult](#). Set this field to true for

[processingInformation.authorizationOptions.ignoreCvResult](#)

[processingInformation.authorizationOptions.ignoreCvResult](#)

[processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionId](#)
[processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason](#)
[processingInformation.authorizationOptions.initiator.storedCredentialUsed](#)
[processingInformation.authorizationOptions.initiator.type](#)
[processingInformation.commerceIndicator](#)
[processingInformation.industryDataType](#) Set this field to `transit`.
[processingInformation.reconciliationId](#)

Related information

- [API Field Reference for the REST API](#)

REST Example: Performing a Merchant-Initiated Authorization for Debt Recovery with a Token

Request

```

{
  "clientReferenceInformation": {
    "transactionId": "29987654321",
    "partner": {
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "processingInformation": {
    "processingInformation.commerceIndicator": "moto",
    "industryDataType": "transit",
    "reconciliationId": "fgssgsgsgsf",
    "authorizationOptions": {
      "debtRecoveryIndicator": "true",
      "ignoreAvsResult": "true",
      "ignoreCvResult": "true",
      "initiator": {
        "type": "merchant",
        "storedCredentialUsed": "true",
        "merchantInitiatedTransaction": {
          "reason": "1",
          "previousTransactionId": "123456766012345"
        }
      }
    }
  },
  "paymentInformation": {
    "card": {
      "expirationMonth": "12",
      "expirationYear": "2031",
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "CD616772D8355EA6E053AF598E0AE794"
    }
  }
}

```

```

    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "currency": "THB"
    }
  }
}
}

```

Response to a Successful Request

```

{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6482309374186627704003/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6482309374186627704003"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6482309374186627704003/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_3",
    "transactionId": "29987654321"
  },
  "id": "6482309374186627704003",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "10.00",
      "currency": "THB"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "accountFeatures": {
      "category": "A",
      "group": "0"
    },
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "CD616772D8355EA6E053AF598E0AE794",
      "state": "ACTIVE"
    }
  },
  "card": {

```

```

    "type": "001"
  }
},
"pointOfSaleInformation": {
  "emv": {
    "tags": "5004564953419F26087C14E9BE1F1065094F07A0000000031010820220009F360203709F0702C0809F2701409"
  }
},
"processorInformation": {
  "systemTraceAuditNumber": "191368",
  "approvalCode": "831000",
  "networktransactionId": "016153570198200",
  "transactionId": "016153570198200",
  "responseCode": "00",
  "avs": {
    "code": "1"
  }
},
"reconciliationId": "fgssgsgsgsfg",
"status": "AUTHORIZED",
"submitTimeUtc": "2022-03-25T17:55:37Z"
}

```

Mass Transit Tap-Initiated Sale for Debt Recovery with a Token

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

Endpoint

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

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

Required Fields for a Tap-Initiated Sale for Debt Recovery with a Token Using the REST API

[clientReferenceInformation.partner.thirdPartyCertificationNumber](#)

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.type](#)

[pointOfSaleInformation.catLevel](#)

[pointOfSaleInformation.entryMode](#)

[pointOfSaleInformation.terminalCapability](#)

[pointOfSaleInformation.terminalId](#)

[pointOfSaleInformation.terminalPinCapability](#)

[processingInformation.authorizationOptions](#). Set this field to `true` or

[processingInformation.authorizationOptions](#). Set this field to `true` or

[processingInformation.capture](#) Set this field to `true`.

[processingInformation.captureOptions.dateToCapture](#)

[processingInformation.commerceIndicator](#) Set this field to `retail`.

[processingInformation.industryDataType](#) Set this field to `transit`.

[processingInformation.reconciliationId](#)

[tokenInformation.jti](#)

Related information

- [API Field Reference for the REST API](#)

REST Example: Performing a Tap-Initiated Sale for Debt Recovery with a Token

Request

```
{
  "clientReferenceInformation": {
    "transactionId": "12987654321",
    "partner": {
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "processingInformation": {
    "industryDataType": "transit",
    "reconciliationId": "dsgfsgsgsfdsgf",
    "captureOptions": {
      "dateToCapture": "0114"
    },
    "capture": "true",
    "processingInformation.commerceIndicator": "retail",
    "authorizationOptions": {
      "debtRecoveryIndicator": "true",
      "deferredAuthIndicator": "true"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",

```

```

    "currency": "THB"
  }
},
"paymentInformation": {
  "card": {
    "type": "001"
  }
},
"tokenInformation": {
  "jti": "a76392f4-cde4-97aa-1111-0242ac14c005"
},
"pointOfSaleInformation": {
  "terminalId": "12345678",
  "catLevel": "2",
  "entryMode": "contactless",
  "terminalCapability": "5",
  "terminalPinCapability": "0"
}
}
}

```

Response to a Successful Request

```

{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/payments/6502821914766725604006/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6502821914766725604006"
    }
  },
  "clientReferenceInformation": {
    "code": "123456",
    "transactionId": "12987654321"
  },
  "id": "6502821914766725604006",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "authorizedAmount": "10.00",
      "currency": "THB"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "accountFeatures": {
      "category": "A",
      "group": "0"
    },
    "tokenizedCard": {

```



```

    "type": "001"
  },
  "card": {
    "type": "001"
  }
},
"pointOfSaleInformation": {
  "emv": {
    "tags": "5004564953419F26087C14E9BE1F1065094F07A0000000031010820220009F360203709F0702C0809F2701409"
  }
},
"processorInformation": {
  "systemTraceAuditNumber": "114880",
  "approvalCode": "831000",
  "networktransactionId": "016153570198200",
  "retrievalReferenceNumber": "210811114880",
  "transactionId": "12987654321",
  "responseCode": "00",
  "avs": {
    "code": "2"
  }
},
"reconciliationId": "dsgfsgsgsfdsgf",
"status": "AUTHORIZED",
"submitTimeUtc": "2022-04-18T11:43:11Z"
}

```

Mass Transit Merchant-Initiated Sale for Debt Recovery with a Token

This section describes how to process a merchant-initiated sale for debt recovery with a token.

Endpoint

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

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

Required Fields for a Merchant-Initiated Sale for Debt Recovery with a Token Using the REST API

[clientReferenceInformation.partner.thirdPartyCertificationNumber](#)

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.cardType](#)

[paymentInformation.card.expirationMonth](#)

[paymentInformation.card.expirationYear](#)

[paymentInformation.instrumentIdentifier.id](#)

[processingInformation.authorizationOptions](#). **Set this field to `true`.**

[processingInformation.authorizationOptions.ignoreAvsResult](#)

[processingInformation.authorizationOptions.ignoreCvResult](#)

[processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction](#).

[previousTransactionId](#)

[processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason](#)

[processingInformation.authorizationOptions.initiator.storedCredentialUsed](#)

[processingInformation.authorizationOptions.initiator.type](#)

[processingInformation.capture](#) **Set this field to `true`.**

[processingInformation.commerceIndicator](#) **Set this field to `moto`.**

[processingInformation.industryDataType](#) **Set this field to `transit`.**

[processingInformation.reconciliationId](#)

Related information

- [API Field Reference for the REST API](#)

REST Example: Performing a Merchant-Initiated Sale for Debt Recovery with a Token

Request

```
{
  "clientReferenceInformation": {
    "transactionId": "26987654321",
    "partner": {
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "processingInformation": {
    "commerceIndicator": "moto",
    "industryDataType": "transit",
    "reconciliationId": "fgsgsgsgsfg",
    "capture": "true",
    "authorizationOptions": {
      "debtRecoveryIndicator": "true",
      "ignoreAvsResult": "true",
      "ignoreCvResult": "true",
      "initiator": {
        "type": "merchant",
        "storedCredentialUsed": "true",
        "merchantInitiatedTransaction": {
```

```

        "reason": "1",
        "previousTransactionId": "123456789012345"
    }
}
},
"paymentInformation": {
    "card": {
        "expirationMonth": "12",
        "expirationYear": "2031",
        "cardType": "001"
    },
    "instrumentIdentifier": {
        "id": "CD616772D8355EA6E053AF598E0AE794"
    }
},
"orderInformation": {
    "amountDetails": {
        "totalAmount": "10.00",
        "currency": "THB"
    }
}
}
}

```

Response to a Successful Request

```

{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/payments/6482305297396608504004/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6482305297396608504004"
    }
  },
  "clientReferenceInformation": {
    "code": "TC50171_3",
    "transactionId": "26987654321"
  },
  "id": "6482305297396608504004",
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "10.00",
      "authorizedAmount": "10.00",
      "currency": "THB"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "accountFeatures": {

```

```

    "category": "A",
    "group": "0"
  },
  "tokenizedCard": {
    "type": "001"
  },
  "instrumentIdentifier": {
    "id": "CD616772D8355EA6E053AF598E0AE794",
    "state": "ACTIVE"
  },
  "card": {
    "type": "001"
  }
},
"pointOfSaleInformation": {
  "emv": {
    "tags": "5004564953419F26087C14E9BE1F1065094F07A0000000031010820220009F360203709F0702C0809F2701409"
  }
},
"processorInformation": {
  "systemTraceAuditNumber": "190723",
  "approvalCode": "831000",
  "networktransactionId": "016153570198200",
  "transactionId": "016153570198200",
  "responseCode": "00",
  "avs": {
    "code": "1"
  }
},
"reconciliationId": "fgssgsgsgsf",
"status": "AUTHORIZED",
"submitTimeUtc": "2022-03-25T17:48:50Z"
}

```

Mass Transit Stand-Alone Credit with a Token

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

A *stand-alone credit* is a credit that is not linked to a capture. There is no time limit for requesting a stand-alone credit.

When a request for a credit is successful, the issuing bank for the payment card takes money out of your merchant bank account and returns it to the customer. It usually takes two to four days for your acquiring bank to transfer funds from your merchant bank account.

Carefully control access to the credit service. Do not request this service directly from your customer interface. Instead, incorporate this service as part of your customer service process.

**Warning**

Follow the preceding guidelines to prevent unauthorized credits.

Endpoint

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

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

Required Fields for a Stand-Alone Credit with a Token Using the REST API

[clientReferenceInformation.partner.thirdPartyCertificationNumber](#)

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

[paymentInformation.card.expirationMonth](#)

[paymentInformation.card.expirationYear](#)

[paymentInformation.card.type](#)

[paymentInformation.instrumentIdentifier.id](#)

Related information

- [API Field Reference for the REST API](#)

REST Example: Performing a Stand-Alone Credit with a Token

Request

```
{
  "clientReferenceInformation": {
    "transactionId": "43987654321",
    "partner": {
      "thirdPartyCertificationNumber": "123456789012"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationMonth": "03",
      "expirationYear": "2031",
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "DCED1B858116F177E053AF598E0AA10A"
    }
  },
  "orderInformation": {
    "amountDetails": {
```

```

    "totalAmount"
  : "200.00",
    "currency": "THB"
  }
}
}

```

Response to a Successful Request

```

{
  "_links": {
    "void": {
      "method": "POST",
      "href": "/pts/v2/credits/6502828338426145304001/voids"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/credits/6502828338426145304001"
    }
  },
  "clientReferenceInformation": {
    "code": "12345678",
    "transactionId": "43987654321"
  },
  "creditAmountDetails": {
    "currency": "THB",
    "creditAmount": "200.00"
  },
  "id": "6502828338426145304001",
  "orderInformation": {
    "amountDetails": {
      "currency": "THB"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "instrumentIdentifier": {
      "id": "DCED1B858116F177E053AF598E0AA10A",
      "state": "ACTIVE"
    },
    "card": {
      "type": "001"
    }
  },
  "reconciliationId": "6502828338426145304001",
  "status": "PENDING",
  "submitTimeUtc": "2022-04-18T11:53:54Z"
}

```

Mass Transit Follow-On Payment Services

You can request these follow-on transactions for mass transit:

- Capture
- Authorization reversal
- Timeout reversal
- Timeout void

Required Fields for Mass Transit Captures

This table provides information about the fields required for mass transit captures.

Captures

REST API Field	Capture	Information/Value
clientReferenceInformation.comments		For this value, see Transaction Types on page 48.
clientReferenceInformation.partner.thirdPartyCertificationNumber		Cybersource provides the value for this field.
clientReferenceInformation.transactionId		—
orderInformation.amountDetails.currency		—
orderInformation.amountDetails.totalAmount		—

Capture an Authorization

This section describes how to process a capture.

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

Endpoint

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

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

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

Required Fields for a Mass Transit Capture

[clientReferenceInformation.comments](#)

[clientReferenceInformation.partner.thirdPartyCertificationNumber](#) The value for this field is provided by Cybersource.

[clientReferenceInformation.transactionId](#)

[orderInformation.amountDetails.currency](#)

[orderInformation.amountDetails.totalAmount](#)

Related Information

- [API field reference guide for the REST API](#)

REST Example: Capture an Authorization

Request

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

Response to a Successful Request

```
{
```



```

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

```

Required Fields for Mass Transit Reversals

This table provides information about the fields required for mass transit reversals.

Reversal Fields

REST API Field	Authorizati on Reversal	Timeout Reversal	Information/Value
clientReferenceInformation.comments			For this value, see Transaction Types on page 48.
clientReferenceInformation.partner.thirdPartyCertificationNumber			Cybersource provides the value for this field.
clientReferenceInformation.transactionId		—	—
orderInformation.amountDetails.currency		—	—

REST API Field	Authorizati on Reversal	Timeout Reversal	Information/Value
orderInformation.amountDetails.totalAmount			—
reversalInformation.reason	—		—

Authorization Reversal

This section describes how to reverse an authorization.

Use the authorization reversal service to reverse an unnecessary or undesired authorization.

Endpoint

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

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

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

Required Fields for a Mass Transit Authorization Reversal

[clientReferenceInformation.comments](#)

[clientReferenceInformation.partner.thirdPartyCertificationNumber](#)

[clientReferenceInformation.transactionId](#)

[reversalInformation.amountDetails.currency](#)

[reversalInformation.amountDetails.totalAmount](#)

Related Information

- [API field reference guide for the REST API](#)

REST Example: Reversing a Mass Transit Authorization

Request

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

```

    "currency": "EUR"
  }
}
}

```

Response to a Successful Request

```

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

```

Timeout Reversal

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

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

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

Required Fields for a Timeout Reversal

[clientReferenceInformation.comments](#) Set this field to `REVERSAL Timeout`.

[clientReferenceInformation.transactionId](#)

[reversalInformation.amountDetails.totalAmount](#)

reversalInformation.reason

Related information

- API Field Reference for the REST API

REST Example: Timeout Reversal

Request

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

Response to a Successful Request

```
{
  "_links": {
    "self": {
      "method": "GET",
      "href": "/pts/v2/reversals/6502854707106431104004"
    }
  },
  "clientReferenceInformation": {
    "code": "1650285470690",
    "transactionId": "78885555"
  },
  "id": "6502854707106431104004",
  "orderInformation": {
    "amountDetails": {
      "currency": "EUR"
    }
  },
  "pointOfSaleInformation": {
    "emv": {
      "tags":
"50004564953419F26087C14E9BE1F1065094F07A0000000031010820220009F360203709F0702C0809F2701409F100706010A"
    }
  },
  "processorInformation": {
    "responseCode": "00"
  },
  "reconciliationId": "6502854707106431104004",
  "reversalAmountDetails": {
    "reversedAmount": "10.00",
    "currency": "EUR"
  }
}
```

```

},
"status": "REVERSED",
"submitTimeUtc": "2022-04-18T12:37:50Z"
}

```

Response to a Decline Request

```

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

```

Required Field for Timeout Voids

This table provides information about the field required for timeout voids.

Timeout Void Field

REST API Field	Timeout Void	Information/Value
clientReferenceInformation.comments		For this value, see Transaction Types on page 48.
clientReferenceInformation.transactionId		Set this field to the unique clientReferenceInformation.transactionId value from the transaction that timed out.

Timeout Void

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

Endpoint

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

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

Required Field for a Timeout Void

[clientReferenceInformation.comments](#)

Set this field to REVERSAL Timeout.

[clientReferenceInformation.transactionId](#)

Related information

- [API Field Reference for the REST API](#)

REST Example: Timeout Void

Request

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

Response to a Successful Request

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

Response to a Declined Request

```
{
  "id": "6502858209346457804004",
  "submitTimeUtc": "2022-04-18T12:43:41Z",
  "status": "INVALID_REQUEST",
  "reason": "INVALID_DATA",
}
```

```
"message": "Declined - One or more fields in the request contains invalid data"  
}
```

Mass Transit Token Management Services

Use the Token Management Service to create, retrieve, and delete tokens for mass transit.

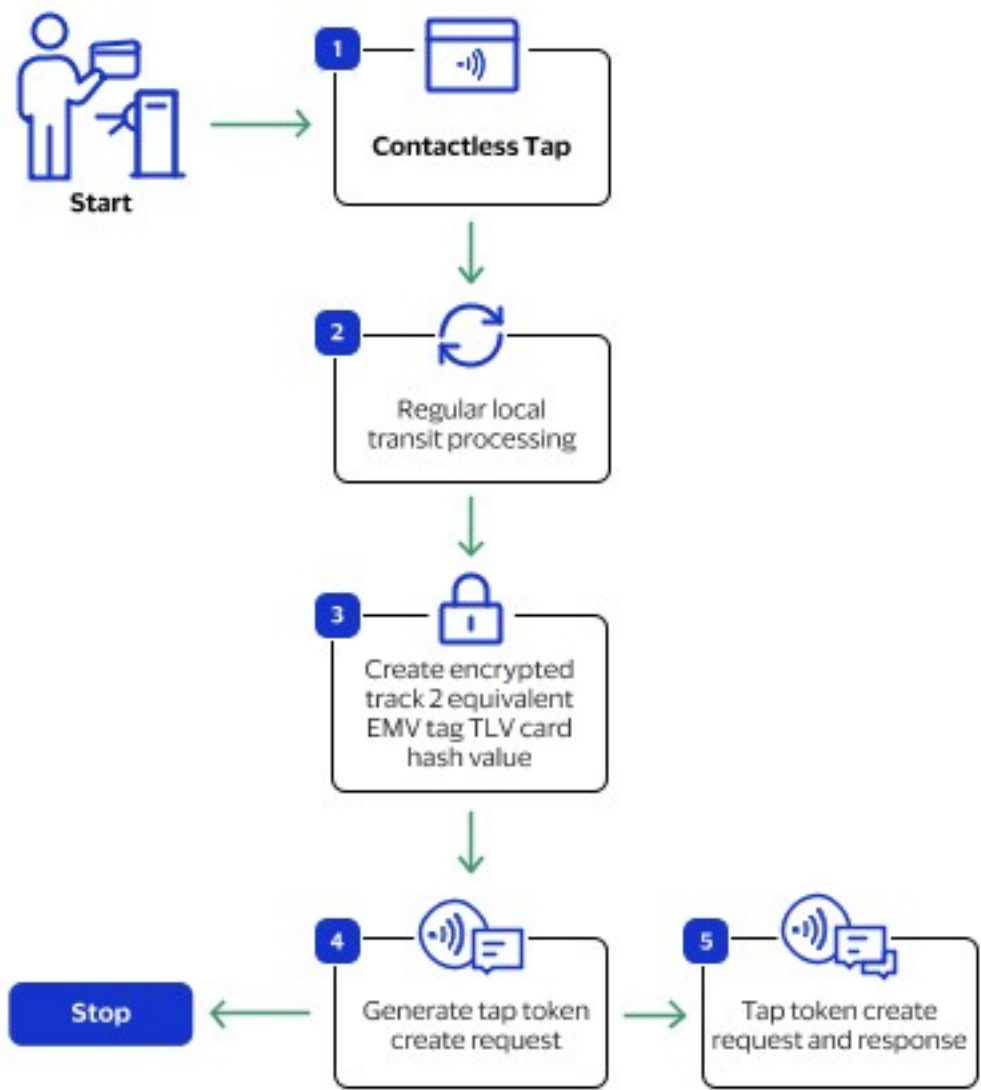
Creating a Token

This section describes how to create tokens directly from the validator or through your back office system.

When the customer taps their card, the validator generates a unique transaction identifier that is used to create these tokens:

- Transient token: tokenized EMV tag data and track2 data. Used as the transaction ID to create the instrument identifier and payment instrument tokens, and as the payment token for the final authorization and sale transactions. You can delete this token after a success transit payment for the travel period. Cybersource stores this token for 7 days.
- Instrument identifier: tokenized card number, used for follow-on payment transactions and BIN lookup.
- Payment instrument: stores the card hash that is used to identify the payment card in the deny list.

This diagram shows the process for creating a token.



Creating a Token

The token creation process begins when the when the cardholder taps a payment card at the fare collection terminal.

1. The cardholder taps their contactless card.
2. Regular local transit processing occurs. See [Mass Transit Transaction Workflows](#) on page 31.
3. The validator encrypts the card data and creates the card hash value.
4. Your system uses the card hash value to create the tokens.
5. You receive the tokens in the response message.

Endpoint

Test: POST <https://apitest.cybersource.com/tms/v2/taps>

Production: POST <https://api.cybersource.com/tms/v2/taps>

Required Fields for Creating a Token Using the REST API

id	Transient token identifier assigned by the contactless terminal.
paymentInformation.card.hash	
paymentInformation.fluidData.descriptor	
paymentInformation.fluidData.encoding	
paymentInformation.fluidData.value	
pointOfSaleInformation.deviceId	
processingInformation.industryDataType	Set this field to <code>transit</code> .

Related information

- [API Field Reference for the REST API](#)

REST Example: Create a Token

Request

```
{
  "id": "a76392f4-cde4-97aa-1111-0252ac14c005",
  "paymentInformation": {
    "card": {
      "hash": "7400A4154369E584BA36CA19B50AAA3F9AE9764B3E87B2B93CF8FBC11112"
    },
    "fluidData": {
      "descriptor": "4649443D454D562E5041594D454E542E415049",
      "encoding": "Hex",
      "value": "DFEE120A8888885140001840000357189141DC75F996D2E930B9ADABE73
5CBE80ECF17A263E94964F07A0000000031010500b4379626572736F7572
63655F2D02456E5F3401015F360102820220008407A00000000310109505
0000000009A032101199C01009F02060000000004009F03060000000000
009F030200019F1006A1B2C3D4E5F69F1A0208409F21031301255F2A0208
409F26080123456789ABCDEF9F2701409F33030000009F360200019F3501
259F3704123456789F390100"
    }
  },
  "pointOfSaleInformation": {
    "deviceId": "FF123457"
  },
  "processingInformation": {
    "industryDataType": "transit"
  }
}
```

Response 202

No body response

Retrieving Transient Token Details

This section describes how to use the token retrieval service to get the transient token details such as the instrument identifier token and payment instrument token. Use the payment instrument token for token management. Use the instrument identifier for these transactions:

- Merchant-initiated debt recovery
- Stand-alone credit

Endpoint

Test: POST <https://apitest.cybersource.com/tms/v2/taps/{id}>

Production: POST <https://api.cybersource.com/tms/v2/taps/{id}>

REST Example: Retrieve Transient Token Details Using the REST API

Request

```
{
}
```

Response 200

```
{
  "id": "a76392f4-cde4-97aa-1111-0252ac14c005",
  "paymentInformation": {
    "card": {
      "expirationMonth": "12",
      "expirationYear": "2030"
    }
  },
  "pointOfSaleInformation": {
    "deviceId": "FF123457",
    "emv": {
      "applicationIdentifier": "A00000000031010",
      "applicationLabel": "Company"
    }
  },
  "processingInformation": {
    "industryDataType": "transit"
  },
  "tokenInformation": {
    "instrumentIdentifier": {
      "id": "CD616772D8355EA6E53AF598E0AE794"
    },
    "paymentInstrument": {
      "id": "DB0875B76F95085CE053AF598E0A6354"
    }
  },
  "_links": {
    "self": {
```

```

    "href": "/tms/v2/taps/a76392f4-cde4-97aa-1111-0252ac14c005"
  },
  "paymentInstrument": {
    "href": "/tms/v1/paymentinstruments/DB0875B76F95085CE053AF598E0A6354"
  },
  "instrumentIdentifier": {
    "href": "/tms/v1/instrumentidentifiers/CD616772D835EA6E53AF598E0AE794"
  }
}
}
}

```

Retrieving an Instrument Identifier Details

This section describes how to retrieve instrument identifier details.

Endpoint

Test: `GET https://apitest.cybersource.com/tms/v1/instrumentidentifiers/{id}`

Production: `GET https://api.cybersource.com/tms/v1/instrumentidentifiers/{id}`

The `id` is the instrument identifier ID that was returned in a token management service response.

```

},
"tokenInformation": {
  "instrumentIdentifier": {
    "id": "CD616772D8355EA6E053AF598E0AE794"
  }
},

```

REST Example: Retrieving Instrument Identifier Details Using the REST API

Request

```

{
}

```

Response 200

```

{
  "_links": {
    "self": {
      "href": "https://apitest.cybersource.com/tms/v1/instrumentidentifiers/CD616772D8355EA6E053AF598E0AE794"
    },
    "paymentInstruments": {
      "href": "https://apitest.cybersource.com/tms/v1/instrumentidentifiers/CD616772D8355EA6E053AF598E0AE794/paymentinstruments"
    }
  },
  "id": "CD616772D8355EA6E053AF598E0AE794",
  "object": "instrumentIdentifier",

```

```

"state": "ACTIVE",
"card": {
  "number": "411111XXXXXXXXXX"
},
"processingInformation": {
  "authorizationOptions": {
    "initiator": {
      "merchantInitiatedTransaction": {
        "previousTransactionId": "016153570198200"
      }
    }
  }
}
}
}
}

```

Retrieving Card Hash Details

This section describes how to use the card hash to retrieve the details, which include the instrument identifier token and payment instrument token.

Endpoint

Test: GET <https://apitest.cybersource.com/tms/v1/paymentinstruments/{id}>

Production: GET <https://api.cybersource.com/tms/v1/paymentinstruments/{id}>

The `id` is the transient token identifier assigned by the contactless terminal.

REST Example: Retrieving Card Hash Details Using the REST API

Request

```
{
}
```

Response 200

```

{
  "_links": {
    "self": {
      "href": "https://apitest.cybersource.com/tms/v1/paymentinstruments/
CD98B27F62A8AECBE053AF598E0AF965"
    }
  },
  "id": "CD98B27F62A8AECBE053AF598E0AF965",
  "object": "paymentInstrument",
  "state": "ACTIVE",
  "type": "cardHash",
  "card": {
    "expirationMonth": "12",
    "expirationYear": "2030",
    "type": "visa",
    "hash": "7400A4154369E584BA36CA19B50AAA3F9AE97FEAE64B3E87B2B93CF8FBC97777"
  }
},

```

```

"metadata": {
  "creator": "tml_tap02"
},
"_embedded": {
  "instrumentIdentifier": {
    "_links": {
      "self": {
        "href": "https://apitest.cybersource.com/tms/v1/instrumentidentifiers/
CD616772D8355EA6E053AF598E0AE794
      },
      "paymentInstruments": {
        "href": "https://apitest.cybersource.com/tms/v1/instrumentidentifiers/
CD616772D8355EA6E053AF598E0AE794/paymentinstruments
      }
    },
    "id": "CD616772D8355EA6E053AF598E0AE794",
    "object": "instrumentIdentifier",
    "state": "ACTIVE",
    "card": {
      "number": "476173XXXXXXXXXX"
    },
    "processingInformation": {
      "authorizationOptions": {
        "initiator": {
          "merchantInitiatedTransaction": {
            "previousTransactionId": "016153570198200"
          }
        }
      }
    },
    "metadata": {
      "creator": "tml_tap01"
    }
  }
}
}
}

```

Retrieving Payment Instrument Details

This section describes how to send a **GET** request to retrieve payment instrument details, which include the card details, card hash, and instrument identifier.

Endpoint

Test: **GET** <https://apitest.cybersource.com/tms/v1/paymentinstruments/{id}>

Production: **GET** <https://api.cybersource.com/tms/v1/paymentinstruments/{id}>

The **id** is the transient token identifier assigned by the contactless terminal.

REST Example: Retrieving Payment Instrument Details Using the REST API

Request

```
{
}
```

Response 200

```
{
  "_links": {
    "self": {
      "href": "https://apitest.cybersource.com/tms/v1/paymentinstruments/
CD98B27F62A8AECBE053AF598E0AF965"
    }
  },
  "id": "CD98B27F62A8AECBE053AF598E0AF965",
  "object": "paymentInstrument",
  "state": "ACTIVE",
  "type": "cardHash",
  "card": {
    "expirationMonth": "12",
    "expirationYear": "2030",
    "type": "visa",
    "hash": "7400A4154369E584BA36CA19B50AAA3F9AE97FEAE64B3E87B2B93CF8FBC97777"
  },
  "_embedded": {
    "instrumentIdentifier": {
      "_links": {
        "self": {
          "href": "https://apitest.cybersource.com/tms/v1/instrumentidentifiers/
CD616772D8355EA6E053AF598E0AE794"
        },
        "paymentInstruments": {
          "href": "https://apitest.cybersource.com/tms/v1/instrumentidentifiers/
CD616772D8355EA6E053AF598E0AE794/paymentinstruments"
        }
      },
      "id": "CD616772D8355EA6E053AF598E0AE794",
      "object": "instrumentIdentifier",
      "state": "ACTIVE",
      "card": {
        "number": "476173XXXXXXXXXX"
      },
      "processingInformation": {
        "authorizationOptions": {
          "initiator": {
            "merchantInitiatedTransaction": {
              "previousTransactionId": "016153570198200"
            }
          }
        }
      }
    }
  }
}
```

```
}
}
```

Deleting an Instrument Identifier

This section describes how to send a **DELETE** request to delete an instrument identifier.

Endpoint

Test: **DELETE** <https://apitest.cybersource.com/tms/v1/instrumentidentifiers/{id}>

Production: **DELETE** <https://api.cybersource.com/tms/v1/instrumentidentifiers/{id}>

The **id** is the instrument identifier ID that was returned in a token management service response.

```
},
  "tokenInformation": {
    "instrumentIdentifier": {
      "id": "CD616772D8355EA6E053AF598E0AE794"
    }
  },
}
```

REST Example: Deleting an Instrument Identifier Using the REST API

Request

```
{
}
```

Response 204

No response body

Response 409

```
{
  "_links": {
    "paymentInstruments": {
      "href": "https://apitest.cybersource.com/tms/v1/instrumentidentifiers/
CD776F4472D6AC69E053AF598E0A30F3/paymentinstruments"
    }
  },
  "errors": [
    {
      "type": "instrumentIdentifierDeletionError",
      "message": "Action cannot be performed as the InstrumentIdentifier is associated with one or more PaymentInstru
    }
  ]
}
```

Response 410: Deleting an Instrument Identifier

```
{
```



```

"errors": [
  {
    "type": "notAvailable",
    "message": "Token not available"
  }
]
}

```

Deleting a Card Hash

This section describes how to send a **DELETE** request to delete a card hash.

Endpoint

Test: **DELETE** <https://apitest.cybersource.com/tms/v1/paymentinstruments/{id}>

Production: **DELETE** <https://api.cybersource.com/tms/v1/paymentinstruments/{id}>

The **id** is the **card.hash** value that was returned in the retrieve a payment instrument response.

```

"card": {
  "expirationMonth": "12",
  "expirationYear": "2030",
  "type": "visa",
  "hash": "7400A4154369E584BA36CA19B50AAA3F9AE97FE93CF8FBC97777"
}

```

REST Example: Deleting a Card Hash

Request

```

{
}

```

Response 204: Deleting a Card Hash

No response body

Response 410: Deleting a Card Hash

```

{
  "errors": [
    {
      "type": "notAvailable",
      "message": "Token not available"
    }
  ]
}

```

Transit Test Cases

Message-Level Validation Test Cases

Case #	Transaction	Card Type	Amount	Comments
Visa AVR and Sale for Aggregated			Transaction	
1	Account verification request (AVR)	Visa	0.00	For Visa, an AVR is performed when the card is first used in the transit system or on a more frequent basis depending on what the PTO/PTP requires.
2	Deferred sale for aggregated transaction	Visa	9900.00	
Visa First Ride		Protection		
3.1	Deferred sale for an aggregated transaction	Visa	9904.00	Response is a decline that is eligible for capture.
3.2	Follow-on capture	Visa	9904.00	Even though the amount exceeds what is allowed for captured under Visa's First Ride Protection rules, proceed with the capture in order to complete this test case.

Case #	Transaction	Card Type	Amount	Comments
Mastercard Authorization and			Capture	
4.1	Authorization for an aggregated transaction	Mastercard	10.00	
4.2	Follow-on capture of an aggregated transaction	Mastercard	9900.00	
Debt Recovery				
5.1	Deferred sale	Mastercard, Visa	9904.00	Response is a decline that is not eligible for capture. Attempt to reclaim debt using MOTO, tap-initiated, merchant-initiated, card-not-present debt recovery.
5.2	MOTO debt recovery	Mastercard, Visa	9601.00	Response is a decline, but it allows you to validate the Debt Recovery payload.
6.2	Tap-initiated debt recovery	Mastercard, Visa	9904.00	Response is a decline, but it allows you to validate the Debt Recovery payload.
7.2	Merchant-initiated debt recovery	Mastercard, Visa	9904.00	Response is a decline, but it allows you to validate the Debt Recovery payload.
8.2	Card-not-present debt recovery with payer authentication	Mastercard, Visa	9904.00	Response is a decline, but it allows you to validate the Debt Recovery payload.

Case #	Transaction	Card Type	Amount	Comments
9.2	Card-not-present debt recovery without payer authentication	Mastercard, Visa	9904.00	Response is a decline, but it allows you to validate the Debt Recovery payload.
Follow-On Transactions				
12.2	Stand-alone credit on test case 02	Visa	20.00	Illustrates your ability to process a credit for an overcharged amount.
13.2	Void on stand-alone credit test case 12.2	Visa	9900.00	Illustrates your ability to void a stand-alone credit that was processed incorrectly.
14	Reversal of test case 02	Visa	9900.00	Illustrates your ability to reverse an authorized amount when the final fare is higher than what was originally authorized. After a reversal, resubmit the correct sale amount.
Transaction Search				
15.1	Deferred sale	Visa	9900.00	Ignore the response in order to simulate a timeout.

Case #	Transaction	Card Type	Amount	Comments
15.2	Transaction Search	—	—	<p>The test case 15.1 should show as successful, and therefore no further action is required. If the transaction was declined, the transaction would be placed on the deny list, and first ride protection or debt recovery process should be initiated.</p>