- Version
  - History
- Introduction
- Audience
- Authentication
- Environments
- URLs
  - Server API
  - WPF
  - Consumer API
  - Tokenization API
  - Genesis KYC Services JSON API
  - Transaction API
  - Smart Routing API
  - Billing Transactions API
- Transactions
  - Invoking a Transaction
  - Card
    - Recurring V2
      - Initial Recurring
      - Subsequent Recurring
      - Non-referenced Subsequent Recurring
      - Recurring for Indian cards
        - How to use managed recurring for Indian cards in Processing API
        - How to use managed recurring for Indian cards in WPF API
    - Recurring Transactions
      - Init Recurring Sale
      - Recurring Sale
      - Recurring for Indian cards
        - How to use managed recurring for Indian cards in Processing API
        - How to use managed recurring for Indian cards in WPF API
    - Authorize
    - Capture
    - Sale
    - Argencard
    - Aura
    - Bancontact
    - Cabal
    - Cencosud
    - Credit (CFT)
    - Elo
    - Naranja
    - Nativa
    - Tarjeta Shopping
  - Non-financial Transactions
    - Account Verification v2
    - Account Verification
  - 3DS Card
    - Authorize 3D
    - Sale 3D
    - Init Recurring Sale 3D
    - Notification for Asynchronous Payments
  - Wallets
    - Neteller
    - PayPal
    - WebMoney
    - Wechat
    - Alipay
    - eZeeWallet
    - eWallet
  - Vouchers
    - CashU
    - Neosurf
    - Paysafecard
  - Online Banking ePayments (oBeP)
    - Banco do Brasil
    - Bancomer
    - Bradesco
    - EPS
    - iDebit
      - Payin
      - Payout
    - iDeal
    - InstaDebit
      - Payin
      - Payout
    - Itau
    - Multibanco
    - MyBank

    - Online Banking
    - P24
        - Bank Codes
    - PayU
    - POLi
    - PSE (Pagos Seguros en Linea)
    - Post Finance
    - RapiPago
    - SafetyPay
    - Santander
    - SDD Init Recurring Sale
    - SDD Recurring Sale
    - SDD Refund
    - SDD Sale
    - Sofort
    - Trustly Sale
    - UPI
    - Webpay
    - Davivienda
- Cash Payments
    - Cash
        - Cash Payment Types
    - Baloto
    - Banco de Occidente
    - Boleto
    - Efecty
    - OXXO
    - Pago Facil
    - PIX
    - Redpagos
- Gift Cards
    - Intersolve
    - Fashioncheque
    - TCS
    - Split Payments
- Invoice Payment Methods
    - Invoice
    - Invoice Capture
- Crypto
    - BitPay Sale
- Payouts
    - Bank Pay-out
        - Bank Names
        - For BRL currency
    - BitPay Payout
    - eZeeCard Payout
    - Payout
    - Money transfer Payout
    - Non-money transfer Payout
    - SCT Payout
    - African Mobile Payout
    - Russian Mobile Payout
    - TransferTo Payout
- Mobile Payments
    - Apple Pay
    - Google Pay
    - African Mobile Sale
    - Russian Mobile Sale
- Reversals
    - Refund
    - Async Refund
    - Void
    - Invoice Refund
    - BitPay Refund
    - Partial Reversal
- PayByLink
    - Workflow
    - Combination with Pay Later functionality
    - Reminders configuration
- Alternative Payment Method External Events
    - Introduction
    - List of external events per alternative payment method
    - Email Notification
    - API Notification
- Advanced risk management with RiskParams
- Credential On File (COF)
- Currency and Amount Handling
- Dynamic Descriptor
- Electronic Commerce Indicator
- Scheme Response Codes
    - Visa
    - Mastercard and Maestro
- Scheme Account Number Indicator Codes
- Manually Reviewed Transactions
- Partial Approvals
- Preauthorizations
    - Preauthorization
        - Visa
        - Mastercard
        - Maestro
    - Incremental Authorize
    - Capture
    - Full Reversal
    - Partial Reversal
- Required vs Optional API params
- Transaction States
- Supported Card Brands
- Document ID Parameter
- Business Attributes
- Recurring Advice
- Tokenized e-commerce
- Customer Identification Parameters
    - Owner
    - Type
    - Subtype
    - Document ID
    - Issuing Country
- Scheme Maximum Settlement Date

- Reconcile

- Single Transaction
    - Preauthorization
- By date range

- Payment Authorizations
  - Payment Authorizations API
  - By date range
- Processed Transactions
  - Processed Transaction API
  - Single Processed Transaction
  - By date or post date range
- Processed Batches
  - Processed Batches API
  - By date or post date range
- Chargebacks
  - Chargeback reversals
  - Chargeback notifications
  - Chargeback API
  - Single Chargeback
  - List of Chargebacks
  - By date range
  - Chargeback types
- Rapid Dispute Resolution
  - Rapid Dispute Resolution API
  - Single Rapid Dispute Resolution
  - List of Rapid Dispute Resolutions
  - By date range
  - Rapid Dispute Resolution types
- Retrieval Requests
  - Retrieval Request notifications
  - Retrieval Request API
  - Single Retrieval Request
  - List of Retrieval Request
  - By date range
- Funding Accounts
  - Funding Accounts API
    - By Post Date
- Fraud reports
  - SAFE/TC40 API
  - Fraud report codes
  - Single SAFE/TC40 report
  - List of SAFE/TC40 report
  - By date range
- Blacklists
  - Invoking a Request
  - Response
- Asynchronous Transactions and Notifications
  - Asynchronous Transactions
  - Notifications
- WPF
  - Workflow
  - Express Checkout
    - Apple Pay
    - Google Pay
  - WPF API
    - URLs
    - Create
    - Notification
    - Reconcile
    - Reconcile by Date range
    - WPF States
    - Errors
  - WPF transaction types
  - WPF Internationalization (i18n)
- 3DSecure
  - V2
    - Introduction
    - Request Params
      - 3DS-Method Params
      - Control Params
      - Recurring Params
    - Authentication Flows
      - Frictionless
        - Diagram
        - Reconcile
        - Notification
      - Frictionless with 3DSecure Method
        - Diagram
        - Reconcile
        - Notification
      - Challenge
        - Diagram
        - Reconcile
        - Notification
      - Challenge with 3DSecure Method
        - Diagram
        - Reconcile
        - Notification
    - Status Reason Code
    - Testing
- Authentication Services
  - Introduction
    - SCA with 3DSecure
    - Alternative services
- Genesis KYC Services
  - General Info

- Create Consumer Registration
- Update Consumer Registration
- Create Transaction
- Update Transaction
- Identity Document Upload
- Identity Document Download
- Make call
- Update call
- Create Verification
  - Supported Document Types
  - Supported Address Types
  - Available Statuses
  - Supported Languages
  - Supported Verification Modes
  - AML Filters
- Verification Status
- Verification Register
- Error Response
- Kyc Service Notification
- CPF check
- CNPJ check

- Genesis Fx Services
  - General Info
  - Get Tiers
  - Get Tier
  - Get Rates
  - Get Rate
  - Search Rate

- Consumers
  - Introduction
  - Consumer API
    - Create consumer
    - Retrieve consumer
    - Update consumer
    - Disable consumer
    - Enable consumer
    - Get consumer cards

- Managed Recurring
  - Introduction
  - How to use managed recurring in Processing API
    - Requests
  - How to use managed recurring in WPF API
    - Requests

- Tokenization
  - Introduction
  - Tokenization API
    - Accepted cardholder parameters
    - Consumer required
    - Tokenize
    - Detokenize
    - Update token
    - Validate Token
    - Delete Token
    - Get card
  - How to tokenize cardholder data in Processing API
    - Create a Consumer
    - Use existing consumer
  - How to use tokens in Processing API
    - Requests
  - How to tokenize cardholder data in WPF API
    - Create a Consumer
    - Use existing consumer
  - How to use tokens in WPF API
  - Supported transaction types
  - Importation of external tokens and card details
    - Input CSV file format
    - Encryption of the CSV file
    - Uploading the encrypted CSV file to remote SFTP server
    - Downloading a response CSV file from the remote SFTP server
    - Response CSV file format

- Transaction API
  - Transaction Card Expiry Date Update API

- APM Services
  - Alipay Register Merchant
  - Introduction
    - Alipay Register Merchant Site info parameters
  - Site info
  - Klarna
  - Introduction
    - Release Remaining Authorization API
    - Resend Invoice API
    - Update Order Items API
    - Update Order Address API
  - Trustly Select Account
  - Introduction
  - Trustly Register Account
  - Introduction
  - TransferTo
  - Introduction
    - Retrieve Payers API

- Errors
  - Error groups table
  - Error codes tables

- Client Integrations

- Client-side encryption
  - Client side
    - Prevent form submit action
    - JavaScript only
  - Server side

- Shopping Carts

- Testing
  - 3DS v2
  - AVS
- Status Page
- Infrastructure and Uptime
- Penetration Testing Warning
- AVS Status Codes
  - Single character code
  - Two-characters code
- CVV Result Codes
- Level 3 Travel Data
  - Travel Types
    - Airline Itinerary Data (AID)
    - Car Rental
    - Hotel Rental
    - Ancillary Charges
    - Miscellaneous Charges
  - Special Cases
    - Travel Authorize (3D) and Capture
    - Visa Refund
  - Allowed Values
    - Car Rental Classes
    - Charge Types
    - Car Rental Extra Charges
    - Hotel Rental Extra Charges
    - Ticket Change Indicators
    - Credit Reason Indicators
- Genesis SCA Services
  - General Info
  - SCA Checker
    - SCA Exemption Values
    - SCA Reason For Not Honoring Exemption Values
    - SCA Exemption Result Values
- Funding Transactions
  - General Info
  - Funding Transaction identifier types
  - Funding Transaction Business Application Identifier Types
  - Funding Transaction Receiver Account types
- Account Name Inquiry
  - Introduction
  - How to use account name inquiry parameters
    - Requests
    - Responses
      - Expected response values for the `*_name` attributes are *M* - Match, *P* - Partial match or *N* - No match
      - Expected response values for the `type` attribute are *P* - primary or *S* - secondary.
- Smart Routing
  - Introduction
  - How Smart Routing works
    - Description of the algorithm
  - How to use Smart Routing
- Installments
  - Fetch Applicable Installments
  - Show Installment Details
- Billing Transactions API
  - General Info
  - Schema and introspection
  - Authorization
    - Authorization error response status codes
  - Billing transactions retrieval
    - Request headers
    - Request filter parameters
    - Request paging and order parameters
    - Response fields
    - Paging response fields
    - Response error codes


---

# Version

---

**API version 49.1**

**Documentation version 49.1**

---

## History

| Version | Date | Name | Description |
|---|---|---|---|
| 1.0 | 2011/06/10 | Emil Petkov | Initial version |
| 1.1 | 2011/11/11 | Emil Petkov | Added optional shipping params section for the shipping address |
| 1.2 | 2012/04/18 | Emil Petkov | Added dynamic descriptor functionality |
| 1.3 | 2012/08/03 | Emil Petkov | Added new transaction type AVS and Account Verification |
| 2.0 | 2013/03/29 | Emil Petkov | Added gaming and MOTO flags and support |
| 2.1 | 2013/04/11 | Emil Petkov | Split Credits with an initial ref transaction and Payouts without a reference transaction |
| 2.2 | 2013/05/26 | Emil Petkov | Added new transaction type InitRecurringSale3D - recurring payments with initial 3D |

| | | | |
|---|---|---|---|
| 2.3 | 2014/01/15 | Emil Petkov | Removed the transaction types DebitSale and IdealSale |
| 2.4 | 2014/01/22 | Emil Petkov | Added authorization code and issuer response code to API, section for issuer response codes |
| 2.5 | 2014/01/29 | Emil Petkov | Moto and gaming flags are now returned in the transaction response if present/marked in the request |
| 2.6 | 2014/02/05 | Emil Petkov | Dynamic descriptor params are now returned in the transaction response if present in the request |
| 2.7 | 2014/02/15 | Emil Petkov | Recurring advices are now returned in the transaction response if received from the issuer |
| 2.8 | 2014/05/01 | Emil Petkov | Added required vs. optional API params |
| 2.9 | 2014/06/16 | Emil Petkov | Currency handling rework and API description (different currency exponents, etc) |
| 3.0 | 2014/08/28 | Emil Petkov | Added risk related APIs - chargebacks, retrieval requests, blacklists |
| 3.1 | 2014/11/25 | Emil Petkov | Added API support for partial approvals |
| 3.2 | 2014/12/03 | Emil Petkov | Added WPF I18N specifics |
| 3.3 | 2014/12/08 | Emil Petkov | Added shopping carts and client integrations list |
| 3.4 | 2014/12/16 | Hristo Tanchev | Now the remote ip can be either a required or optional API param |
| 3.5 | 2015/02/22 | Dimitar Kostov | Added API for eZeeWallet and PayByVoucher via Yeepay |
| 3.6 | 2015/02/23 | Hristo Tanchev | Added API for CashU and Paysafecard |
| 3.7 | 2015/03/13 | Dimitar Kostov | Added API for Sofort |
| 3.8 | 2015/03/18 | Hristo Tanchev | Added API for PPRO |
| 3.9 | 2015/04/15 | Emil Petkov | New WPF API and WPF payment methods |
| 4.0 | 2015/05/14 | Hristo Tanchev | Added API for Neteller and ABN iDEAL |
| 4.1 | 2015/06/15 | Dimitar Kostov | Added WPF custom attributes - bin, tail |
| 4.2 | 2015/08/10 | Tsvetelina Borisova | Added WPF custom attributes - default, expiration date |
| 4.3 | 2015/08/12 | Vladimir Korichkov | Added API for WebMoney and POLi |
| 4.4 | 2015/08/19 | Tsvetelina Borisova | Added fraud related API for TC40/SAFE (fraud reports) |
| 4.5 | 2015/08/27 | Emil Petkov | 3D attributes xid and cavv are now not re- quired in the MPI sync attempted only workflow, only eci is |
| 4.6 | 2015/09/02 | Emil Petkov | New transaction type PayByVoucher Sale for purchasing vouchers via credit cards. Reworked the PayByVouchers section |
| 4.7 | 2015/09/04 | Emil Petkov | Added penetration testing warning for merchants |
| 4.8 | 2015/09/06 | Emil Petkov | Reconcile API now works with ARN and transaction ID in addition to unique ID |
| 4.9 | 2015/09/11 | Hristo Tanchev | Added Electronic Commerce Indicator to notifications |
| 5.0 | 2015/09/30 | Vladimir Korichkov | Added API for INPay |
| 5.1 | 2015/12/09 | Pepa Simeonova | Added event parameter to notifications for fraud transactions. |
| 5.2 | 2016/02/02 | Hristo Tanchev | Added API for P24 |
| 5.3 | 2016/02/16 | Tsvetelina Borisova | Added ARN in reconcile response if available. |
| 5.4 | 2016/03/18 | Tsvetelina Borisova | Added page for Manually Reviewed Transactions |
| 5.5 | 2016/03/17 | Tsvetelina Borisova | Added API for SDD. |
| 5.6 | 2016/03/28 | Emil Petkov | Extended PayByVouchers processing and WPF APIs with redeem type and card type |
| 5.7 | 2016/04/08 | Emil Petkov | Added info for statuspage.io, uptime and infrastructure, and new shopping carts |
| 5.8 | 2016/05/11 | Tsvetelina Borisova | Change the example for chargeback API - amount is returned in minor currency unit |
| 5.9 | 2016/05/22 | Emil Petkov | Now birth date API param is required only for Visa initial transactions with Financial Service MCCs (e.g. 6012) |
| 6.0 | 2016/06/30 | Pepa Simeonova | Added dynamic descriptor params to WPF payments. |
| 6.1 | 2016/07/20 | Stefan Slaveykov | Now usage can be either a required or optional API param |
| 6.2 | 2016/07/21 | Tsvetelina Borisova | Update documentation for chargebacks API |
| 6.3 | 2016/12/08 | George Naydenov | Added API for Trustly |
| 6.4 | 2017/01/20 | Plamen Terziev | Added AVS Codes |
| 6.5 | 2017/02/16 | Nikolay Petrov | Added API for PayPal Express Checkout |
| 6.6 | 2017/03/20 | Iskar Enev | Added API for Citadel |
| 6.7 | 2017/03/30 | Stanislav Mihailov | Added API for Instadebit/iDebit |
| 6.8 | 2017/03/31 | Tsvetelina Borisova | Added API for SCT Payout |
| 6.9 | 2017/05/12 | Nikolay Petrov | Added reference transaction unique id to the Reconcile API response |
| 7.0 | 2017/05/13 | Nikolay Petrov | Added API for Earthport |
| 7.1 | 2017/05/15 | Stanislav Mihailov | Added API for Wechat |
| 7.2 | 2017/05/30 | George Naydenov | Added API for Alipay |
| 7.3 | 2017/06/05 | Hristo Tanchev | Added API for PaySec |
| 7.4 | 2017/06/06 | Hristo Tenev | Added card brand and card number tags in reconcile response. |
| 7.5 | 2017/06/27 | Lukasz Wojcik | Added lifetime for WPF Payments. |
| 7.6 | 2017/07/06 | Nikolay Petrov | Removed AVS transaction type. AVS response code and text are returned in the transactions responses when present and supported or in the notifications for payments with asynchronous workflow. |
| 7.7 | 2017/07/06 | Iskar Enev | Added APM External Events. |
| 7.8 | 2017/08/11 | Samuil Goranov | Added support for optional shipping address params in Processing and WPF APIs. |
| 7.9 | 2017/08/17 | Lukasz Wojcik | Remove remote ip parameter for transactions with reference. It will be copied from reference transaction |
| 8.0 | 2017/08/18 | Lukasz Wojcik | Added invalid transaction types for amount parameter to WPF response. |
| 8.1 | 2017/08/18 | Emil Kirilov | Added search option by import date in Chargeback, Retrieval request and Fraud APIs. |

| | | | |
|---|---|---|---|
| 8.2 | 2017/09/05 | George Naydenov | Added API for RPN Payment |
| 8.3 | 2017/10/13 | Ventsislav Dimitrov | Added API for FashionCheque gift card and split payments. |
| 8.4 | 2017/10/13 | Pepa Simeonova | Added API for Intersolve gift card and split payments. |
| 8.5 | 2017/10/13 | Stanislav Mihailov | Added API for TCS gift card and split payments. |
| 8.6 | 2017/10/23 | Iskar Enev | Added Authentication Services and iSignThis. |
| 8.7 | 2017/11/02 | George Naydenov | Added API for RPN Payout |
| 8.8 | 2017/11/24 | Hristo Tenev | Added API for Paycommerce |
| 8.9 | 2017/11/25 | George Naydenov | API update with new sections Card, 3DS Card, Common. Removing older payment methods. |
| 9.0 | 2017/12/18 | George Naydenov | Added API for Neosurf. |
| 9.1 | 2018/01/02 | George Naydenov | Removed API for Inpay, ABN iDeal, Teleingreso and PayByVoucher. |
| 9.2 | 2018/01/03 | Iskar Enev & Nikolay Petrov | Added Tokenization service. |
| 9.3 | 2018/01/12 | Hristo Tanchev | Added API for Klarna. |
| 9.4 | 2018/01/18 | Stefan Petrov | Added ARN in WPF reconcile response if available. |
| 9.5 | 2018/01/26 | George Naydenov | Added API for Astropay Direct, Pago Facil, Link, Carulla, Davivienda. |
| 9.6 | 2018/02/22 | George Kostov & Hristo Tenev | Added API for Genesis KYC Services. |
| 9.7 | 2018/02/22 | Emil Kirilov | Added API for PSE. |
| 9.8 | 2018/02/26 | Stefan Petrov | Added API for RapiPago, Webpay, Banco de Chile. |
| 9.9 | 2018/03/01 | Stefan Petrov | Added API for Surtimax, Efecty, Cabal, Cencosud, Hipercard, Elo, Aura, Itau, Bradesco, Tarjeta Shopping, BBVA Bancomer, Boleto, Redpagos, Emprese De Energia, GiroPay, InstantTransfer, Multibanco. |
| 10.0 | 2018/03/02 | George Naydenov | Added API for OXXO, Argencard, Naranja, Nativa, Cartao Mercado Livre, Astropay Card, Banamex, Santander, Santander Cash, Zimpler PayU. |
| 10.1 | 2018/03/02 | Emil Kirilov | Added API for Baloto, Banco de Occidente, Banco do Brasil. |
| 10.2 | 2018/04/10 | Stefan Petrov | Added API for Entercash. |
| 10.3 | 2018/05/08 | Hristo Tenev | Added API for eZeeWallet Payout. |
| 10.4 | 2018/05/31 | Maya Nedyalkova | Added 3D MasterCard test cards. |
| 10.5 | 2018/05/31 | George Naydenov | Added API for QQPay. |
| 10.6 | 2018/06/22 | Hristo Tanchev | Added API for Credential on File (COF). |
| 10.7 | 2018/07/05 | George Naydenov | Removed Paysec and added Online Banking OBeP. |
| 10.8 | 2018/07/18 | Stanislav Mihailov | Added the API param issuer_oct_enabled to Visa-based Account Verification transactions, to allow merchant to verify if issuer supports OCTs for the given PAN |
| 10.9 | 2018/07/27 | Yordan Pulov | Added Money Transfer support to Payouts. |
| 11.0 | 2018/08/01 | George Naydenov | Removed Paysec Payout and added Bank Payout. |
| 11.1 | 2018/08/02 | Stanislav Mihailov | Remove API for Citadel |
| 11.2 | 2018/08/31 | Hristo Tenev | Added API for BitPay. |
| 11.3 | 2018/08/31 | Maya Nedyalkova | Add rc_code and rc_description in notification parameters. |
| 11.4 | 2018/10/10 | Ralitsa Borisova | Remove Paycommerce, RPN Payment, Link, Davivienda, Banco de Chile, Cartao Mercado Livre documentation. |
| 11.5 | 2018/10/11 | Emil Petkov | Introduced debt repayments - now birth date API param is required also for Mastercard/- Maestro initial transactions with Financial Service MCCs (e.g. 6012), where merchant is UK-based, transaction is domestic (with UK-based bin), and card type is DEBIT. |
| 11.6 | 2018/11/02 | Yordan Pulov and Stanislav Mihailov | Introduced Travel layer and Added Level 3 Travel API. |
| 11.7 | 2018/11/21 | Hristo Tenev | Added Payout support for BitPay |
| 11.8 | 2018/12/05 | Stefan Petrov | Added API for eZeeCard Payout |
| 11.9 | 2018/12/05 | Nikolay Petrov | Added Pay Later support to the WPF. Introduced Reminders module. |
| 12.0 | 2019/01/07 | Yordan Pulov | Changed Contract Merchant Category Codes for Level 3 Visa Car and Hotel Rental Transactions. |
| 12.1 | 2019/01/31 | Aleksandar Krastev | Introduced Consumer API. Extended Tokenization API to require a consumer. Extended WPF and Processing APIs to create consumers and tokenize card details in one step. WPF API can use saved cards to make payments. |
| 12.2 | 2019/02/18 | Yordan Pulov | Extend the Online Banking bank codes and add payment type. |
| 12.3 | 2019/02/26 | Stefan Petrov | Added API for Tola payments. |
| 12.4 | 2019/03/13 | Stanislav Mihailov | Added support for new non-money transfer payout types. |
| 12.5 | 2019/03/19 | Nikolay Valchanov | Added cardholder and expiration dates params to reconcile APIs. |
| 12.6 | 2019/03/19 | Vladislav Yakimov | Added retrieve endpoint for the Consumer API. |
| 12.7 | 2019/03/22 | Ventsislav Dimitrov | Added support for Preauthorizations. |
| 12.8 | 2019/03/28 | Nikolay Petrov | Added support for importation of external tokens and card details. |
| 12.9 | 2019/04/01 | Nikolay Petrov & Yasen Angelov | Added support for Pay by Link functionality. |
| 13.0 | 2019/04/09 | Hristo Tenev | Added API for transaction card expiry date update. |
| 13.1 | 2019/04/11 | Stanislav Mihailov | Added API support for purchase of cryptocurrency transactions. |
| 13.2 | 2019/04/12 | George Naydenov | Removed QQPay transaction type. |
| 13.3 | 2019/04/15 | Nikolay Valchanov | Added usage and description params to WPF reconcile API. |
| 13.4 | 2019/04/22 | Nikolay Valchanov | Added optional bic param to iDeal transactions. |
| 13.5 | 2019/05/31 | George Naydenov | Added Genesis Fx Services. |
| 13.6 | 2019/06/07 | Rumen Milushev | Added WPF API Reconcile by_date. |
| 13.7 | 2019/06/14 | Yordan Pulov | Extended the Online Banking bank codes and Bank Payout banks. |
| 13.8 | 2019/07/02 | Aleksandar Krastev | Added Tokenization API get masked card details for token. |

| | | | |
|---|---|---|---|
| 13.9 | 2019/07/04 | Vladislav Yakimov | Added API for TransferTo Payout and TransferTo Payers retrieve. |
| 14.0 | 2019/07/09 | Stefan Petrov | Extend the Online Banking bank codes. |
| 14.1 | 2019/07/11 | Pepa Simeonova | Added new money-transfer types. |
| 14.2 | 2019/07/16 | Hristo Tenev | Added API support for Business attributes. |
| 14.3 | 2019/07/22 | Pepa Simeonova | Added source_of_funds as an optional API param for OCT types (Credit, Payout) |
| 14.4 | 2019/08/01 | Stefan Petrov | Removed API for Astropay Card, Astropay Direct, Hipercard, Carulla, Emprese de Energia, Surtimax. |
| 14.5 | 2019/08/05 | Hristo Tanchev | Extended Processing and WPF APIs support for FX (Forex). |
| 14.6 | 2019/08/05 | Martin Lazarov | Added reversible amount in Preauthorization reconcile response. |
| 14.7 | 2019/08/06 | Rumen Milushev | Added MOTO flag to WPF transaction types: Authorize, Authorize3D, Sale, Sale3D. |
| 14.8 | 2019/08/07 | Yasen Angelov | Added the new optional param reminder_language to the WPF API. |
| 14.9 | 2019/08/13 | Stefan Petrov | Rebranding African Mobile payments. |
| 15.0 | 2019/08/21 | Nikolay Petrov | Added new status 'represented' in notifications for Processing and WPF APIs. |
| 15.1 | 2019/08/22 | Stefan Petrov | Added beneficiary params to refund transaction. |
| 15.2 | 2019/09/03 | Stefan Petrov | Use sync workflow for Neosurf transaction. |
| 15.3 | 2019/09/13 | Hristo Tanchev | Extended transaction response with transaction ID form card schemes. |
| 15.4 | 2019/09/13 | Yordan Pulov | Added Genesis SCA Checker service API. |
| 15.5 | 2019/09/16 | Nikolay Petrov | Added support for Account Verification to the WPF API. |
| 15.6 | 2019/09/17 | Ventsislav Dimitrov | Extended the supported Merchant Category Codes for Visa Preauthorization |
| 15.7 | 2019/09/18 | Aleksandar Krastev | Added Processed Transaction API for Card Present and Card Not Present transactions. |
| 15.8 | 2019/09/20 | Yordan Pulov | Extended Genesis SCA Checker service API documentation. |
| 15.9 | 2019/09/20 | Yordan Pulov | Added new fields and changed the endpoint for sending TransferToAPI requests. |
| 16.0 | 2019/09/24 | Rumen Milushev | Added 'agency_name' to Hotel Rental Travel attributes. |
| 16.1 | 2019/09/25 | Aleksandar Krastev | Changed filter flags of Processed Transaction API by_date and by_post_date endpoints. |
| 16.2 | 2019/10/09 | Aleksandar Krastev | Extended Chargeback API to return more attributes; Added flags filtering by origin and type of processing. |
| 16.3 | 2019/10/10 | Pepa Simeonova | Added new MPI parameters and SCA parameters. |
| 16.4 | 2019/10/16 | Aleksandar Krastev | Extended Retrieval Request API to return more attributes; Added flags filtering by origin and type of processing. |
| 16.5 | 2019/10/17 | Martin Lazarov | Extended the supported Merchant Category Codes for Visa Preauthorization. |
| 16.6 | 2019/10/21 | Yordan Pulov | Added support for UATP Travel. |
| 16.7 | 2019/10/24 | Rumen Milushev | Transferred Hotel Rentals, Car Rentals and Cruise Lines from Travel Level 3 data to Business Attributes. |
| 16.8 | 2019/10/24 | Martin Lazarov | Added Reference Transaction Unique ID to notification parameters for the reference-based transactions. |
| 16.9 | 2019/10/31 | Martin Lazarov | Added Capture tolerance for Mastercard and Maestro Preauthorizations. |
| 17.0 | 2019/11/05 | Yordan Pulov | Added new fields for TransferTo Payout. |
| 17.1 | 2019/11/05 | Nikolay Petrov | Added new supported currencies and banks for Bank Payout. |
| 17.2 | 2019/11/15 | Ventsislav Dimitrov | Added API support for purchasing Mastercard and Maestro crypto-currencies. |
| 17.3 | 2019/11/15 | Nikolay Petrov | Added optional bank params to the refund transaction. |
| 17.4 | 2019/11/19 | Stefan Petrov | Added Processed Batches API. |
| 17.5 | 2019/11/20 | Pepa Simeonova | Added sub_merchant_id as an optional dynamic descriptor param. |
| 17.6 | 2019/11/20 | Rumen Milushev | Added optional time extensions to the Processing Reconcile and WPF Reconcile APIs. |
| 17.7 | 2019/11/21 | Nikolay Petrov | Added support for Online Banking Unified Payment Interface (UPI) payment type. |
| 17.8 | 2019/11/27 | Stefan Petrov | Added Batch and Deposit Slip Numbers to Processed Transaction API response. |
| 17.9 | 2019/11/28 | Stefan Petrov | Removed API for Entercash and Banamex. |
| 18.0 | 2019/12/03 | Nikolay Petrov | Made BIC param optional for SddSale and SddInitRecurringSale transactions. |
| 18.1 | 2019/12/05 | Nikolay Petrov | Added supported bank codes for Online Banking that can be used with Netbanking payment type. |
| 18.2 | 2019/12/12 | George Naydenov | Added Trustly Select Account API. Extended parameters for Trustly Sale and Bank Pay-out. |
| 18.3 | 2019/12/18 | Rumen Milushev | Added descriptions for Airlines and Travel agencies in Business Attributes. |
| 18.4 | 2019/12/20 | George Naydenov | Added birth_date as conditionally required API param for Trustly Sale and Bank Pay-out. |
| 18.5 | 2020/01/08 | Hristo Tenev | Added Funding Account API for Card Present and Card Not Present transactions. |
| 18.6 | 2020/01/21 | Sridhar Belagod | Added Trustly Register Account API. |
| 18.7 | 2020/01/29 | Sridhar Belagod | Marked birth_date as optional param for Trustly and changed description. |
| 18.8 | 2020/01/29 | Ralitsa Borisova | Added Finnish language as part of the platform internationalization |
| 18.9 | 2020/01/30 | Sridhar Belagod | Updated supported countries for Trustly Sale. |
| 19.0 | 2020/02/04 | Dmitri Lihachev | Added API for Apple Pay. |
| 19.1 | 2020/02/05 | Sridhar Belagod | Removed unique_id param from Trustly RegisterAccount API. Updated birth_date format and extended the example to include country-specific format |
| 19.2 | 2020/02/05 | Stefan Petrov | Exposed querying and reconciling by ARN for Fraud reports, Chargebacks and Transactions. |
| 19.3 | 2020/02/21 | Ralitsa Borisova | Updated description for Trustly Sale. |
| 19.4 | 2020/02/26 | Rumen Milushev | Added batch_slip_number and deposit_slip_number as optional search params for ProcessedTransactions and ProcessedBatches. |
| 19.5 | 2020/02/26 | Rumen Milushev | Added type and card_number response params to Processed Transactions API response. |
| 19.6 | 2020/02/28 | Svilen Siderov | Updated MyBank supported countries for PPRO. |
| 19.7 | 2020/03/11 | Stanislav Mihailov | Added Payment Authorizations API for Card Present and Card Not Present authorizations. |

| | | | |
|---|---|---|---|
| 19.8 | 2020/03/18 | Nikolay Petrov | Added captured flag and capturable_amount param to Reconcile API response for authorizations. |
| 19.9 | 2020/03/18 | Stefan Petrov | Updated supported countries for PaysafeCard. |
| 20.0 | 2020/03/19 | Stefan Petrov | Removed France and United Kingdom from the supported countries for Sofort. |
| 20.1 | 2020/03/18 | Teodor Nikolov | Updated amount fields descriptions. |
| 20.2 | 2020/03/23 | George Naydenov | Added list of supported clearinghouses for Trustly register account call. |
| 20.3 | 2020/03/23 | Rumen Milushev | Added merchant_number, merchant_reference_transaction and capture_method to Chargebacks API response. |
| 20.4 | 2020/03/27 | Rumen Milushev | Unified response param to 'merchant_transaction_reference' in Payment Transactions and Chargebacks API. |
| 20.5 | 2020/04/03 | Ventsislav Dimitrov | Added asynchronous 3DSv2 support for 3D transaction types |
| 20.6 | 2020/04/03 | Svilen Siderov | Added card type, card subtype and card issuing bank to the Processing and WPF Reconcile API responses |
| 20.7 | 2020/04/08 | Stanislav Mihailov | Extended Risk data APIs (TC40/SAFE, Chargebacks, Retrieval Requests) to return a list of records by ARN or by transaction unique ID if the new list mode is enabled |
| 20.8 | 2020/04/15 | Stanislav Mihailov | Updated amount field for Card Present, Card Not Present and External payment authorizations. |
| 20.9 | 2020/04/15 | Martin Lazarov | Added details about Async Refund transaction type. |
| 21.0 | 2020/04/15 | George Naydenov | Added clearing houses list of supported IBANs and account numbers for Trustly Register Account |
| 21.1 | 2020/04/20 | Ventsislav Dimitrov | Changed the Mastercard test card numbers for card enrolled and card not participating in the 3DSv1. |
| 21.2 | 2020/04/28 | Stanislav Mihailov | Made gaming flag optional for purchase of VISA cryptocurrency transactions. |
| 21.3 | 2020/04/28 | Ventsislav Dimitrov | Added additional required recurring params for asynchronous InitRecurringSale3d using 3DSv2 authentication protocol. |
| 21.4 | 2020/05/05 | Ventsislav Dimitrov | Added more advanced merchant sequence flow diagrams for 3DSv1 and 3DSv2 authentication protocols using the Processing API. |
| 21.5 | 2020/05/05 | Ralitsa Borisova | Added Norwegian, Danish and Swedish language as part of the platform internationalization. |
| 21.6 | 2020/05/11 | Svilen Siderov | Added bank account number and bank identifier code to the Processing and WPF Reconcile API responses. |
| 21.7 | 2020/05/15 | Ventsislav Dimitrov | Added more detailed request examples for **synchronous** 3DSecure Visa and MasterCard transactions through the **3DSv1** and **3DSv2** authentication protocol. |
| 21.8 | 2020/05/15 | Ventsislav Dimitrov | Marked `xid` as out of scope for **synchronous** 3DSecure transactions using the **3DSv2** authentication protocol. |
| 21.9 | 2020/05/15 | Ventsislav Dimitrov | Extended the 3DSv2 documentation, authentication flow diagrams, and marked `usage` as required param when processing asynchronous 3DSecure transaction through 3DSv2 authentication protocol. |
| 22.0 | 2020/05/19 | Martin Lazarov | Updated the documentation related to Alipay transaction type, its transaction amount limits and currencies. |
| 22.1 | 2020/06/01 | Mario Chankov | Updated the description related to POLi payment. |
| 22.2 | 2020/06/01 | Svilen Siderov | Added Alipay Register Merchant API. |
| 22.3 | 2020/06/09 | Svilen Siderov | Added auth_start_date and auth_end_date to Payment Authorizations API request. |
| 22.4 | 2020/06/24 | Martin Lazarov | Added `authorization_code` to the API and WPF notification of various transaction types. |
| 22.5 | 2020/06/25 | Martin Lazarov | Added `retrieval_reference_number` to: API/WPF notifications, API/WPF response and API/WPF reconciliation response. |
| 22.6 | 2020/07/01 | Martin Lazarov | Removed **UPI** from Online banking supported payment types. Added new transaction type UPI. |
| 22.7 | 2020/07/07 | Nikolay Petrov | Added support for **AliPay QR** payment type in the Online Banking. |
| 22.8 | 2020/07/10 | Mario Chankov | Removed **Zimpler** and **Santander Cash**. Changes to supported countries for **MyBank**, **Safetypay** and **Santander** |
| 22.9 | 2020/07/17 | Ventsislav Dimitrov | Changed the 3DSv2-Method submission from HTTP GET to HTTP POST with signature mechanism, extended 3DSv2 authentication flow diagrams. |
| 23.0 | 2020/07/17 | Ventsislav Dimitrov | Added API endpoint for handling continuation after 3DSv2-Method submission. |
| 23.1 | 2020/07/17 | Ventsislav Dimitrov | Added `threeds_method_callback_url` as optional request param to 3DS transaction types in asynchronous workflow. |
| 23.2 | 2020/07/17 | Ventsislav Dimitrov | Added 3DS response attributes to the Processing reconciliation and notification for 3DS transaction types in asynchronous workflow. |
| 23.3 | 2020/07/18 | Ventsislav Dimitrov | Dropped the synchronous 3DSv2-Method workflow handling and authentication flow sequence diagrams. |
| 23.4 | 2020/07/20 | Alexey Kuznetsov | Renamed Klarna related transaction to Invoice. Also, migrate Klarna items to transaction items and introduce a new payment type for different types of invoices. |
| 23.5 | 2020/07/21 | Stefan Petrov | Added new **pending_hold** status for async transactions where finalized by user, but final update from provider not yet received (Sofort, etc) |
| 23.6 | 2020/07/22 | Stanislav Mihailov | Updated the documentation for TC40/SAFE APIs (fraud reports) when searching by date. |
| 23.7 | 2020/07/22 | Martin Lazarov | Added details regarding `partially_reversed` payment transaction state. |
| 23.8 | 2020/07/22 | Tsvetan Tsvetanov | Changed Sofort transaction parameter name from `bank_account_number` to `iban` |
| 23.9 | 2020/08/04 | Filipp Pirozhkov | Added new tokenization errors and client-side encryption errors. |
| 24.0 | 2020/08/05 | Vladimir Nudelman | Added `return_pending_url` as an optional param to the WPF API |
| 24.1 | 2020/08/05 | Stefan Petrov | Made **voucher_number** for Neosurf transactions conditionally required. |
| 24.2 | 2020/08/06 | Vladimir Nudelman | Added `cvv_result_code` as an optional parameter in the notifications, transaction responses, and reconciliation responses for both WPF and Processing APIs. |
| 24.3 | 2020/08/17 | Tsvetan Tsvetanov | Added `return_success_url_target` as optional request param to Trustly Sale via Processing API and WPF. |
| 24.4 | 2020/08/24 | Pepa Simeonova | Added Funds Transfer BAI and removed MCC restrictions for money transfer payouts. |
| 24.5 | 2020/08/26 | Martin Lazarov | Added new TransferTo transfer type: `C2C` (Consumer to Consumer) as well as new conditionally required sender parameters, a table with supported destination countries and currencies, and a link to Retrieve Payers API. |
| 24.6 | 2020/08/26 | Martin Lazarov | Removed `NB` as available bank code for INR Online Banking transactions as well as the corresponding Netbanking-related notes. |
| 24.7 | 2020/09/02 | Eduard Gataullin | Removed `currency` attribute from Recurring Sale transaction params, as recurring series work only with the same currency as the init recurring and cannot be changed. |
| 24.8 | 2020/09/09 | Martin Lazarov | Added new conditionally required attributes for TransferTo payouts for the B2B scenario - `document_reference_number` and `purpose_of_remittance`. |
| 24.9 | 2020/09/16 | Alexey Kuznetsov | Added API for Secure Invoice payments. |
| 25.0 | 2020/09/16 | Martin Lazarov | Renamed the following TransferTo payout attributes: `sender_lastname`, `sender_firstname`, `ifs_code` to: `sender_last_name`, `sender_first_name` and `indian_financial_system_code`. |
| 25.1 | 2020/09/16 | George Naydenov | Added API for PostFinance. |
| 25.2 | 2020/09/29 | Martin Lazarov | Clarified the descriptions of the following TransferTo payout attributes: `account_type`, `country`, `msisdn` and `sender_msisdn` |
| 25.3 | 2020/10/02 | Maya Nedyalkova | Added capability to AVS response attributes. |
| 25.4 | 2020/10/21 | Ventsislav Dimitrov | Added field format restrictions to the billing and shipping address for async 3DS transactions that are using the 3DSv2 authentication protocol. |

| | | | |
|---|---|---|---|
| 25.5 | 2020/10/27 | Ventsislav Dimitrov | Added international customer phone field format notes for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| 25.6 | 2020/10/27 | Ventsislav Dimitrov | Added card holder field format restrictions for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| 25.7 | 2020/10/28 | Kuznetsov Alexey | Added API for e-wallet solutions. |
| 25.8 | 2020/10/28 | Ventsislav Dimitrov | Added `state` field format notes for the billing and shipping attributes of async 3DS transactions that are using the 3DSv2 authentication protocol as per the EMVCo spec for 3DSv2.1. |
| 25.9 | 2020/10/28 | Ventsislav Dimitrov | Added hints for building the signatures in the 3DSv2 scope of 3DS transactions that require 3DS-Method submission. |
| 26.0 | 2020/11/04 | Ralitsa Borisova | Extend MCC range for Airlines segment in Business Attributes. |
| 26.1 | 2020/11/04 | Maya Nedyalkova | Added funds status attribute to the Reconcile API responses. |
| 26.2 | 2020/11/13 | Pepa Simeonova & Ventsislav Dimitrov | Added support for 3DSv2 authentication protocol to the Web Payment Form. |
| 26.3 | 2020/11/16 | Pepa Simeonova & Ventsislav Dimitrov | Added additional 3DS attributes to the WPF Notification and WPF reconcile API response. |
| 26.4 | 2020/11/18 | Ventsislav Dimitrov | Changed the additional `purchase` and `recurring` params of the 3DSv2 authentication protocol to **optional** for 3DS transactions through the processing API. |
| 26.5 | 2020/11/24 | Martin Lazarov | Update the description of `payment_type` for Online Banking transactions. |
| 26.6 | 2020/11/24 | George Naydenov | Extend Online Banking banks list and supported currencies. |
| 26.7 | 2020/12/01 | Mario Chankov | Removed Bancolombia and DirectDebit as supported bank codes for ARS and COP currencies under Online Banking. |
| 26.8 | 2020/12/01 | Stanislav Mihailov | Added `scheme_settlement_date` to Processing API and Processing API/WPF API reconciliation responses. Extended the `scheme_transaction_identifier` description and added it to WPF API reconciliation response. |
| 26.9 | 2020/12/03 | George Naydenov | Removed national_id attribute for Online Banking. |
| 27.0 | 2020/12/08 | Martin Lazarov | Added support for `sca_params` for Authorize and Sale transaction types and updated the description for the `exemption` field. |
| 27.1 | 2020/12/08 | Stanislav Mihailov | Added `credential_on_file_settlement_date` attribute to **Authorize** and **Sale** payment transaction types. Extended the **Credential On File** section with a description for `merchant_unscheduled` **COF** type. |
| 27.2 | 2020/12/08 | Vladimir Nudelman | Added support for `auth_network_outage` exemption to **Authorize** and **Sale** transactions. |
| 27.3 | 2020/12/15 | Ventsislav Dimitrov | Added information for `low_risk` SCA exemption to sync/async 3DSv2 transactions and extended the transaction example requests. |
| 27.4 | 2020/12/15 | Ventsislav Dimitrov | Changed the `usage` param to **optional** for async 3DS transactions through the 3DSv2 authentication protocol. |
| 27.5 | 2020/12/16 | Maya Nedyalkova | Removed API for **QIWI** and **InstantTransfer**. |
| 27.6 | 2020/12/30 | Ventsislav Dimitrov | Extended the validation and description of the **browser** request parameters in the scope of 3DS transactions through the 3DSv2 authentication protocol. |
| 27.7 | 2021/01/06 | Martin Lazarov | Added an example **EEA** MasterCard card number for simulating the `low-risk` exemption request in synchronous 3DSv2 workflow. |
| 27.8 | 2021/01/26 | Martin Lazarov | Extended the description of the `color_depth` attribute as part of the required async 3DSv2 browser device channel attributes within the 3DSv2 authentication protocol. |
| 27.9 | 2021/01/26 | Teodor Nikolov | Removed API for Trustly withdrawal. |
| 28.0 | 2021/01/28 | Stefan Petrov | Added bank codes for P24. |
| 28.1 | 2021/01/28 | Eduard Gataullin | Added verification and verification status KYC endpoints. |
| 28.2 | 2020/02/04 | Ventsislav Dimitrov | Added support for numeric chars in the browser language subtag for transactions with 3DSv2 authentication protocol. |
| 28.3 | 2021/02/05 | Pavel Abolmasov | Added `return_pending_url` as optional param for APMs via Processing API. |
| 28.4 | 2021/02/12 | Maya Nedyalkova | Added description for the `bic` field to **iDeal**. Included warning for iframes usage. |
| 28.5 | 2021/02/16 | Ventsislav Dimitrov | Extended the list with the originated IPs for asynchronous payment notifications. |
| 28.6 | 2021/02/12 | Eduard Gataullin | Added background checks support to verification KYC endpoints. |
| 28.7 | 2021/02/18 | Stanislav Mihailov | Added an example **EEA** Visa card number, CAVV and ECI for simulating the `low-risk` exemption request in synchronous 3DSv2 workflow. |
| 28.8 | 2021/03/12 | Evgeny Zhdanov | Added `funds_status` and `account_holder` attributes to WPF reconcile, notification, and to reconcile params. |
| 28.9 | 2021/03/17 | Pepa Simeonova | Added `sca_preference` to WPF API params. |
| 29.0 | 2021/03/18 | Martin Lazarov | Added support for zero-value amounts with the card-based transaction types - **Sale**, **Sale3D**, **Authorize**, **Authorize3D**, **Init Recurring Sale**, **Init Recurring Sale3D**. |
| 29.1 | 2021/03/24 | Martin Lazarov | Added a notice for amount restrictions of the **Partial Reversal** transactions. |
| 29.2 | 2021/04/29 | Eduard Gataullin | Added support for an optional `reference_id` parameter to verification KYC endpoints. |
| 29.3 | 2021/05/11 | Martin Lazarov | Added `payment_type` to the list of supported business attributes. |
| 29.4 | 2021/05/14 | Atanas Zlatarev | Removed `Spain` from the supported countries of **MyBank** transaction. |
| 29.5 | 2021/05/19 | Maya Nedyalkova | Added support for Revolut to **iDeal** transactions. |
| 29.6 | 2021/05/19 | Simeon Angelov | Extended supported currencies for Neosurf. |
| 29.7 | 2021/06/22 | Hristo Tanchev | Extended **Init Recurring** with `scheme_settlement_date` and `scheme_transaction_identifier` attributes. |
| 29.8 | 2021/07/09 | Nikola Yurukov | Updated **WPF** transaction types. |
| 29.9 | 2021/07/15 | Ventsislav Dimitrov | Added additional originating IPs for the asynchronous payment notifications on the Staging environment. |
| 30.0 | 2021/07/21 | Mario Chankov | Removed **Moneyou** from the list of available issuers for the **iDeal** payment method. |
| 30.1 | 2021/07/22 | Rumen Milushev | Added Russian `mobile sale` and `mobile payout` transaction types. |
| 30.2 | 2021/08/03 | George Naydenov | Extended the description for online banking and transaction types that are going to be deprecated. |
| 30.3 | 2021/08/06 | Ventsislav Dimitrov | Extended the description note with the allowed characters of the `card_holder` request param for **Payout** with Visa or MasterCard gambling transactions regarding winnings. |
| 30.4 | 2021/08/16 | Mario Chankov | Added Moldova (MD) to the list of supported countries for the **Paysafecard** transaction type. |
| 30.5 | 2021/08/16 | Pepa Simeonova | Added `sender_birth_date` to money-transfer payout params. |
| 30.6 | 2021/08/18 | Simeon Angelov | Extended the list with supported currencies for **Bitpay Sale** transaction type. |
| 30.7 | 2021/08/23 | Simeon Angelov | Updated the list with supported countries for **Bitpay** transaction types. |
| 30.8 | 2021/09/03 | Ventsislav Dimitrov | Added additional originating IPs for the asynchronous payment notifications on the Production environment. |
| 30.9 | 2021/09/09 | Eduard Gataullin | Made the `country` parameter optional and added support for `address_backside_proof_required` and `expiry_date` as optional parameters for the verification KYC endpoint. |

| | | | |
|---|---|---|---|
| 31.0 | 2021/09/14 | Svilen Siderov | Added API for Google Pay. |
| 31.1 | 2021/09/27 | Ventsislav Dimitrov | Added a warning note how to avoid Cross-origin resource sharing issues during 3DS-Method-Continue submission in the 3DSv2 authentication protocol. |
| 31.2 | 2021/09/28 | Svilen Siderov | Removed API for Earthport. |
| 31.3 | 2021/10/05 | Pepa Simeonova | Added scheme response code and recurring advice in the notifications, transaction responses, and reconciliation responses for both WPF and Processing APIs. |
| 31.4 | 2021/10/12 | Ralitsa Tsonova | Clarified the descriptions for `amount`, `currency` and `chargeback_amount`, `chargeback_currency` in Chargeback API section. |
| 31.5 | 2021/10/15 | Nikolay Petrov | Added optional `purpose_of_payment` parameter for Visa OCTs. |
| 31.6 | 2021/10/26 | Simeon Angelov | Added support for **Low Risk** and **Low Value** SCA exemptions to the WPF API. |
| 31.7 | 2021/11/01 | George Naydenov | Added SCA reasons for not honoring exemption. |
| 31.8 | 2021/11/09 | Svilen Siderov | Added a new payment type for Google Pay API. |
| 31.9 | 2021/11/24 | Teodor Nikolov | Added additional gateways support for the eWallet transaction type with a restricted list of providers. |
| 32.0 | 2021/12/08 | Hristo Tanchev | Added MOTO flag to WPF transaction types: InitRecurringSale and InitRecurringSale3D. |
| 32.1 | 2021/12/09 | George Naydenov | Added Interac Combined Pay-in bank code for Online Banking transaction |
| 32.2 | 2021/12/14 | Blagoy Vangelov | Added new transaction type PIX. |
| 32.3 | 2021/12/17 | Hristo Tanchev | Added support for managed recurring. |
| 32.4 | 2021/12/21 | Simeon Angelov | Added **ECI** to the 3DS attributes in the transaction responses and reconciliation responses for both WPF and Processing APIs. |
| 32.5 | 2021/12/21 | Slavcho Savov | Added support for Shopware and Magento 2.x EE, ECE in the Shopping Carts section. Removed the deprecated Magento 1.x from the supported shopping cart plugins. |
| 32.6 | 2021/12/14 | Svilen Siderov | Added API for **PayPal** transaction type. |
| 32.7 | 2022/01/10 | Boris Kolev | Updated supported banks list for bank payouts - extended banks list for CLP currency. |
| 32.8 | 2022/01/11 | Georgi Naydenov | Updated supported banks list for bank payouts - extended banks list for CAD currency. |
| 32.9 | 2021/01/13 | Teodor Nikolov | Added `report_date` to the Fraud Report API response. |
| 33.0 | 2022/01/13 | Svilen Siderov | Replaced API for **PaypalExpress** with new **express** payment type for **PayPal** API. |
| 33.1 | 2022/02/02 | Ventsislav Dimitrov | Add support for ApplePay on Web Payment Form's express checkout page. |
| 33.2 | 2022/02/03 | Yordan Pulov | Extended KYC verification creation request with optional params. |
| 33.3 | 2022/02/09 | Boris Kolev | Added search by **Report Date** for Fraud Report API. |
| 33.4 | 2022/02/09 | Muhammad Moawaz Ayub | Added API for retrieving Rapid Dispute Resolutions. |
| 33.5 | 2022/02/10 | Yuliyan Dudin | Added `scheme_transaction_identifier` and `scheme_settlement_date` to the API and WPF notifications. |
| 33.6 | 2022/02/15 | Boris Kolev | Changed `recurring` payment type to `init_recurring_sale` for Mobile-Tokenized transactions like **Apple Pay** and **Google Pay**. |
| 33.7 | 2022/02/10 | Evgeny Zhdanov | Extended Bank Pay-out id_card_number with link. Added more bank names for BRL. |
| 33.8 | 2022/02/22 | Teodor Nikolov | Added card issuing country to the Processing and WPF Reconcile API responses. |
| 33.9 | 2022/02/24 | Boris Kolev | Changed `payment_type` to `payment_subtype` for Mobile-Tokenized transactions like **Apple Pay** and **Google Pay**. |
| 34.0 | 2022/02/28 | Mario Chankov | Changed the `bic` and `iban` request parameters for **Giropay** to optional. |
| 34.1 | 2022/03/01 | Ventsislav Dimitrov | Added support for payment subtype `sale` to Apple Pay Processing and WPF API. |
| 34.2 | 2022/03/04 | Blagoy Vangelov | Removed consumer_reference for **PIX** transaction type. The first_name and last_name in billing_address become required for **PIX**. |
| 34.3 | 2022/03/07 | Simeon Angelov | Added 3DS authentication status reason code to: API/WPF notifications, API/WPF response and API/WPF reconciliation response. |
| 34.4 | 2022/03/02 | Milen Matev | Added information for the different environments in a new **Environments** section. |
| 34.5 | 2022/03/17 | Ivan Kolev | Added new chargeback response parameters: `chargeback_account_amount`, `chargeback_account_currency`, `merchant_funding_amount`, `merchant_funding_currency` |
| 34.6 | 2022/03/17 | Boris Kolev | Changed the 3DS-Method status callback from synchronous to asynchronous for all 3DSv2 authentication flows. |
| 34.7 | 2022/03/17 | Boris Kolev | Extended the 3DSv2 authentication flows with conflict response in case of duplicated 3DS-Method continue is requested. |
| 34.8 | 2022/03/22 | Ivan Kolev | Extended the WPF Notification and Async Notification with additional optional parameters that can be requested by the merchant. |
| 34.9 | 2022/03/22 | Svilen Siderov | Added `web_payment_form_id` parameter to WPF create request parameters. |
| 35.0 | 2022/04/04 | Ilya Rogozin | Added **Tokenized e-Commerce** section and introduced `scheme_tokenized` attribute to Processing API for transactions with scheme tokenization support. |
| 35.1 | 2022/04/07 | Ivan Kolev | Added `document_expiry_date` to KYC verification status response. |
| 35.2 | 2022/04/07 | Ivan Kolev | Changed KYC requests domains and endpoints. |
| 35.3 | 2022/04/11 | Boris Kolev | Added `service_provider_name` as a request param to the Money Transfer Payouts. |
| 35.4 | 2022/04/21 | Boris Kolev | Added support for cryptocurrency Visa OCT transactions (**Credit** and **Payout**). |
| 35.5 | 2022/04/27 | Boris Kolev | Updated the **MYR** currency bank codes for `online_banking`. |
| 35.6 | 2022/04/28 | Pepa Simeonova | Added `customer_identification` to Visa OCT transactions (**Credit** and **Payout**) for Brazil and Qatar. |
| 35.7 | 2022/05/10 | Preslav Nedev | Added `bank_code` and `payment_type` to the Online Banking transaction response. |
| 35.8 | 2022/05/16 | Blagoy Vangelov | Added `transaction_id` attribute to Processed Transaction API response for `card not present` payment requests only. |
| 35.9 | 2022/05/18 | Boris Kolev | Updated the **MYR** currency bank codes for `online_banking`. |
| 36.0 | 2022/05/31 | Muhammad Moawaz Ayub | Removed MCC restriction for crypto with Visa. |
| 36.1 | 2022/05/31 | Ivan Kolev | Added optional sub-parameters: `allow_offline` and `allow_online` to both `document` and `face` parameters to enrich the flexibility of the kyc verification request. |
| 36.2 | 2022/07/07 | Simeon Angelov | Added an asynchronous authentication workflow handling for GooglePay transactions thru the 3DS authentication protocol for PAN-ONLY transactions inside the European Economic Area (EEA). |
| 36.3 | 2022/07/07 | Boris Kolev | Added **BCT** bank code to **EUR** currency for `online_banking`. |
| 36.4 | 2022/06/28 | Ivan Kolev | Changed the description of signature in KYC service notification to indicate that it is now being generated by using the API login instead of the API password. |
| 36.5 | 2022/07/13 | Evgeny Zhdanov | Added bank_payout_verification_digit attribute to Bank Payout and barcode, digitable_line, ticket_expiry_date attribute to Boleto. |
| 36.6 | 2022/08/02 | Preslav Nedev | Added business attributes to **Trustly Sale** transactions. |
| 36.7 | 2022/08/24 | Atanas Zlatarev | Fixed discrepancies in description for **iDeal**, **Sofort** and asynchronous transaction notifications. |

| | | | |
|---|---|---|---|
| 36.8 | 2022/08/30 | Boris Kolev | Updated the **SGD** currency bank codes for `online_banking`. |
| 36.9 | 2022/10/06 | Evgeny Zhdanov | Updated **bank_account_type** attribute values related to Itau for Bank Pay-out. |
| 37.0 | 2022/10/05 | Boris Kolev | Added `recurring_category` as an optional request parameter in **InitialRecurringSale** and **InitialRecurringSale3d** transactions for both API and WPF requests. |
| 37.1 | 2022/10/07 | Boris Kolev | Added `sca_exemption_result` to API and WPF transaction responses, reconciliation responses, and notifications. |
| 37.2 | 2022/10/17 | Teodor Nikolov | Removed API for `TrustPay`. |
| 37.3 | 2022/11/04 | Maya Nedyalkova | Updated the list of supported banks for BRL currency. |
| 37.4 | 2022/11/07 | Boris Kolev | Decommissioned **3DSv1** authentication protocol. |
| 37.5 | 2022/11/16 | Preslav Nedev | Added **WPF States** section and linked status field of WPF transactions to it. |
| 37.6 | 2022/11/17 | George Naydenov | Added new recurring types for `sale` and `authorize` and their 3DS variants. Added deprecation notice to the initial and subsequent recurring transaction types. |
| 37.7 | 2022/12/20 | Pepa Simeonova | Added managed recurring params for recurring transactions with Indian cards. |
| 37.8 | 2022/12/21 | Preslav Nedev | Added additional Visa specific authentication status reason codes. |
| 37.9 | 2022/12/22 | Simeon Angelov | Added Dynamic Descriptor params to **GooglePay** and **ApplePay** transactions. |
| 38.0 | 2023/01/03 | Teodor Nikolov | Added currency as an additional parameter to **Processing** and **WPF** notifications. |
| 38.1 | 2023/01/20 | Martin Lazarov | Added `capture_method` to the Processed Transactions API response. |
| 38.2 | 2023/01/25 | Muhammad Moawaz Ayub | Added ACS transaction ID and 3DS challenge indicator as optional params for the 3DS transactions in synchronous workflow. |
| 38.3 | 2023/02/28 | Mladen Rusev | Added company_name and mandate_reference as optional params for SddSale transactions. |
| 38.4 | 2023/03/02 | Atanas Naydenov | Added additional values to Recurring Advice table. |
| 38.5 | 2023/03/22 | Imran Zahoor | Updated Bank-payouts list of banks for BRL currency. |
| 38.6 | 2023/03/29 | Georgi Naydenov | Added subsequent recurring type for Authorize transactions. |
| 38.7 | 2023/04/04 | Imran Zahoor | Added a verification reference id endpoint for KYC to allow registration of reference ids in Genesis. |
| 38.8 | 2023/04/07 | Imran Zahoor | Added a new optional parameter: `check_duplicate_attribute` to the KYC verification request to allow enabling of duplicate detection service. |
| 38.9 | 2023/04/12 | Muhammad Moawaz Ayub | Updated descriptions for VISA AVS codes. |
| 39.0 | 2023/04/20 | Umair Aziz | Extended the list with Recurring advice codes. |
| 39.1 | 2023/04/24 | Pepa Simeonova | Add `user_category` for `online_banking` and `upi`. |
| 39.2 | 2023/04/25 | George Naydenov | Added Recurring V2. |
| 39.3 | 2023/04/26 | Muhammad Moawaz Ayub | Extended the AVS status codes and response processors description. |
| 39.4 | 2023/04/28 | Umair Aziz | Changed the default authorization timeframe and removed MCC restrictions for Visa pre-authorizations and incrementals. |
| 39.5 | 2023/05/03 | Blagoy Vangelov | Added **BL** bank code to **PLN** currency for `online_banking` |
| 39.6 | 2023/05/10 | Ralitsa Galabova | Added Account Verification V2. |
| 39.7 | 2023/05/15 | Blagoy Vangelov | Renamed **BL** bank code to **BLK** |
| 39.8 | 2023/05/16 | Preslav Nedev | Updated documentation on **Verification Status** response for `Shufti Pro`. |
| 39.9 | 2023/05/26 | Imran Zahoor | Added a new parameter **bank_code** for Bank Pay-out transactions with BRL currency. |
| 40.0 | 2023/05/26 | Dimitar Natskin | Updated the Shopping Carts section and deprecated Shopware 5.x plugin. |
| 40.1 | 2023/06/13 | Imran Zahoor | Renamed the parameter `check_duplicate_attribute` of the KYC verification request to `check_duplicate_request`. |
| 40.2 | 2023/06/14 | Ilya Rogozin | Added possibility for customizing the signature algorithm in the Processing Notifications. |
| 40.3 | 2023/06/19 | Umair Aziz | Added missing voucher number to example requests for Neosurf transaction. |
| 40.4 | 2023/06/21 | Evgeny Zhdanov | Added payment_type as conditionally required param to **BankPayout** transactions. |
| 40.5 | 2023/06/22 | Pepa Simeonova | Added additional parameters to the dynamic descriptors. |
| 40.6 | 2023/06/27 | Boris Kolev | Changed the `merchant_zip_code` to be a required parameter for VISA OCT transactions with Australian or Canadian card bins. |
| 40.7 | 2023/07/04 | Teodor Nikolov | Marked first and last name of the billing address as optional params for PIX transactions. |
| 40.8 | 2023/07/04 | Svilen Siderov | Updated description for the additional dynamic descriptors parameters. |
| 40.9 | 2023/07/04 | Muhammad Moawaz Ayub | Added `return_success_url` and `return_failure_url` as conditionally required parameters for Neosurf transactions. |
| 41.0 | 2023/07/11 | Boris Kolev | Updated the **THB** and **VDN** currency bank codes for `online_banking`. |
| 41.1 | 2023/07/05 | Mladen Rusev | Changed bank codes for banks handling IDR currency. |
| 41.2 | 2023/07/20 | Imran Zahoor | Updated description of **bank_code** and **bank_name** for Bank Pay-out transactions with BRL currency. |
| 41.3 | 2023/07/20 | Mladen Rusev | Marked the consumer State in Tokenization as optional for US, CA and CN countries. |
| 41.4 | 2023/07/21 | Muhammad Moawaz Ayub | Added `auth_network_outage` to the list of available exemption types. |
| 41.5 | 2023/08/02 | Teodor Nikolov | Marked country of the billing address as optional param for PIX transactions. |
| 41.6 | 2023/08/17 | Muhammad Moawaz Ayub | Added Funding Transaction support. |
| 41.7 | 2023/08/17 | Boris Kolev | Removed unused **THB** currency bank code for `online_banking`. |
| 41.8 | 2023/09/07 | Svilen Siderov | Added new parameters for `bank_payout` transaction. |
| 41.9 | 2023/09/14 | Simeon Angelov | Added MXN currency support and additional account details response params for Online Banking transactions. |
| 42.0 | 2023/09/21 | Ilya Rogozin | Extended the asynchronous 3DSv2 behaviour for handling invalid submission of 3DS-Method-Continue API calls. |
| 42.1 | 2023/09/28 | Aleksandar Krastev | Added `return_success_url` and `return_failure_url` as conditionally required parameters for SddSale and SddInitRecurringSale transactions. |
| 42.2 | 2023/09/29 | Boris Kolev | Added more testing cards numbers for the `3DSv2` testing section |
| 42.3 | 2023/10/06 | Svilen Siderov | Added support on Processing and WPF APIs for Account Name Inquiries |
| 42.4 | 2023/10/20 | Dimitar Natskin | Added Genesis Ruby SDK. |
| 42.5 | 2023/10/26 | Aleksandar Krastev | Added `return_pending_url` as conditionally required parameter for SddSale and SddInitRecurringSale transactions. |
| 42.6 | 2023/10/26 | Umair Aziz | Extended the example responses for Single Reconciliation and async 3DSv2 flows. |

| | | | |
|---|---|---|---|
| 42.7 | 2023/11/02 | Svilen Siderov | Updated **bank_account_type** attribute values for Bank Payout. |
| 42.8 | 2023/11/15 | Preslav Nedev | Added `customer_id` as required parameter for **Paysafecard** transactions. |
| 42.9 | 2023/11/20 | Pepa Simeonova | Added new recurring advice codes. |
| 43.0 | 2023/11/29 | Blagoy Vangelov | Added new status 'representment reversed' in notifications for Processing and WPF APIs, and a new RepresentmentReversal transaction type. |
| 43.1 | 2023/12/04 | Yordan Pulov, Svilen Siderov, Martin Lazarov | Introduced new Smart Routing API layer allowing for simpler and more efficient gateway integration. |
| 43.2 | 2023/12/08 | Hristo Tanchev | Added support for non-referenced subsequent recurring transactions. |
| 43.3 | 2023/12/11 | Aleksandar Krastev | Added Rapid Dispute Resolution Reversal transaction type to API. |
| 43.4 | 2023/12/13 | Artur Veletskiy | Added new Billing Transactions API |
| 43.5 | 2023/12/14 | Blagoy Vangelov | Added an optional `merchant_website` parameter to the eZeeWallet transaction type for both Processing API and WPF API |
| 43.6 | 2023/12/14 | Blagoy Vangelov | Rename "Blik One Click" payment method to "BLIK" for Online banking. |
| 43.7 | 2024/02/07 | Kaloyan Mirchev | Added CPF/CNPJ checks for Genesis KYC Service. |
| 43.8 | 2024/02/15 | Muhammad Moawaz Ayub | Corrected response params for the Rapid Dispute Resolution. |
| 43.9 | 2024/02/20 | Artur Veletskiy | Updated date only format for Billing Transactions API response. |
| 44.0 | 2024/02/22 | Dimitar Natskin | Added Spree Commerce Shopping Cart plugin. |
| 44.1 | 2024/02/22 | Svilen Siderov | Added Neighborhood as address parameter for processing API and WPF. |
| 44.2 | 2024/02/23 | Aleksandar Krastev | Added **TRL** bank code to **EUR** and **GBP** currencies for `online_banking`. |
| 44.3 | 2024/02/23 | Mladen Rusev | Added `return_pending_url` parameter to PIX documentation. |
| 44.4 | 2024/02/27 | Blagoy Vangelov | `merchant_website` is now required for the `eZeeWallet` transaction type for both Processing API and WPF API. |
| 44.5 | 2024/02/29 | Simeon Angelov | Changed the `purpose_of_payment` to required API param for Visa OCT transactions with recipients in Argentina, Bangladesh, Egypt, India, Chile and Colombia. |
| 44.6 | 2024/03/01 | Atanas Naydenov | Added Scheme response codes and removed Issuer response codes. |
| 44.7 | 2024/03/06 | Kaloyan Mirchev | Added descriptions for CPF/CNPJ checks API response. |
| 44.8 | 2024/03/07 | Muhammad Moawaz Ayub | Added additional MCCs to the list for own-account Funding Transactions. |
| 44.9 | 2024/03/20 | Vladislav Vodenicharski | Extended lifetime validity for Web Payment Form payments to 3 months |
| 45.0 | 2024/03/11 | Blagoy Vangelov | Added `birth_date`, `incorporation_date`, `gender`, `martial_status`, `sender_occupation`, `mothers_name`, `nationality`, `country_of_origin`, `birth_city`, `birth_state`, `company_type`, `company_activity` optional parameters to `PIX` transaction type. |
| 45.1 | 2024/04/08 | Martin Lazarov | Added new recurring advice code for Sanctions Scoring Service. |
| 45.2 | 2024/04/09 | Boris Kolev | Added a new optional request parameter - `protocol_sub_version` to synchronous 3DS transactions. Also, added the 3DS protocol sub-version in reconciliation responses and notifications for Processing and WPF APIs. |
| 45.3 | 2024/06/03 | Ilya Rogozin | Added a new `scheme_transaction_link_id` attribute for the Processing and WPF API responses including reconciliation responses and notifications. |
| 45.4 | 2024/08/13 | Simeon Angelov | Added a list with supported countries for GooglePay transactions. |
| 45.5 | 2024/08/16 | Ilya Rogozin | Added a new `payment_account_reference` attribute for the Processing and WPF API responses, reconciliation responses and notifications. |
| 45.6 | 2024/08/28 | Ilya Rogozin | Extended the list of payment type values for the Online Banking transaction. Added description for Cash transactions. |
| 45.7 | 2024/09/04 | Lachezar Tsvetkov | Added optional `background_checks`.`filters` attribute for KYC Create Verification. |
| 45.8 | 2024/09/05 | Simeon Angelov | Added 0-amount support for GooglePay and ApplePay transactions. |
| 45.9 | 2024/09/09 | Atanas Naydenov | Updated scheme response codes Visa - 1A and Mastercard - 65. |
| 46.0 | 2024/09/12 | Ilya Rogozin | Added extended initial recurring type support to GooglePay and ApplePay transactions. |
| 46.1 | 2024/10/02 | Lachezar Tsvetkov | Changed the `document_supported_types` and `redirect_url` parameters in KYC Create Verification *optional* to allow for synchronous checks. |
| 46.2 | 2024/10/08 | Muhammad Moawaz Ayub | Added VISA Funding Transaction support. |
| 46.3 | 2024/10/09 | Muhammad Moawaz Ayub | Added region availability for Account Name Inquiry Service Functionality. |
| 46.4 | 2024/10/15 | Muhammad Moawaz Ayub | Extended the list with recurring advice codes. |
| 46.5 | 2024/10/16 | Svilen Siderov | Added additional dynamic descriptor parameters related to acceptor geographic coordinates. |
| 46.6 | 2024/10/16 | Atanas Naydenov | Remove API for Giropay transaction type, as it is shut down after June 30 2024. |
| 46.7 | 2024/10/16 | Boris Kolev | Added a new `scheme_max_settlement_date` attribute for the Processing and WPF API responses including reconciliation responses and notifications. Also, the Scheme Max Settlement Date is taken into account and might be prioritized during Capture transactions. |
| 46.8 | 2024/10/18 | Muhammad Moawaz Ayub | Added support for liquid assets in OCT and AFT transactions. |
| 46.9 | 2024/10/18 | Simeon Angelov | Updated supported countries for MyBank transaction type. |
| 47.0 | 2024/10/21 | Lachezar Tsvetkov | Added a new optional parameter `background_checks`.`match_score` in KYC Create Verification. |
| 47.1 | 2024/10/22 | Svilen Siderov | Extended Money transfer Payout with the `credential_on_file_transaction_identifier` parameter. |
| 47.2 | 2024/10/22 | Ilya Rogozin | Added a new attribute `scheme_authentication_data_quality` to the Processing and WPF API responses, as well as reconciliation responses and notifications. |
| 47.3 | 2024/10/23 | Mladen Rusev | Described response CSV file format part of the Importation of external tokens and card details workflow. |
| 47.4 | 2024/10/29 | Mladen Rusev | Added support for AUD currency and PID bank code for Online Banking transaction. |
| 47.5 | 2024/11/15 | Yordan Pulov | Added support of installments. |
| 47.6 | 2024/11/15 | Ilya Rogozin | Added new attributes `scheme_pan_indicator` and `scheme_pan_tail` to the Processing and WPF API responses, as well as reconciliation responses and notifications. |
| 47.7 | 2024/11/21 | Milen Matev | Extended FX service API response examples. |
| 47.8 | 2024/12/05 | Kaloyan Mirchev | Extend Bank Pay-Out request optional parameters. |
| 47.9 | 2024/12/09 | Muhammad Moawaz Ayub | Added Account Funding Transaction support on the Web Payment Form. |
| 48.0 | 2024/12/20 | Zdravko Chorlev | Added initial recurring support to the SCA Checker service API. |
| 48.1 | 2024/12/20 | Artur Veletskiy | Extended Billing Transactions API with additional request and response parameters. |
| 48.2 | 2025/01/09 | Ilya Rogozin | Added support for the NZD currency and the POLI bank code for Online Banking transactions. |
| 48.3 | 2025/01/14 | Muhammad Moawaz Ayub | Added AFT support for Apple Pay and Google Pay transactions. |

| | | | |
|---|---|---|---|
| 48.4 | 2025/01/14 | Svilen Siderov | Extended AVS Status codes with single character codes description. |
| 48.5 | 2025/01/17 | Artur Veletskiy | Changed the request and response params in the Billing Transactions API. |
| 48.6 | 2025/01/22 | Dimitar Natskin | Removed osCommerce plugin from the supported Shopping Carts. |
| 48.7 | 2025/01/22 | Dimitar Natskin | Revised the list with supported Shopping cart plugins. |
| 48.8 | 2025/02/11 | Ilya Rogozin | Added optional beneficiary params to PIX transaction type for the Processing API. |
| 48.9 | 2025/02/11 | Kaloyan Mirchev | Extended Bank Payout API parameters. |
| 49.0 | 2025/02/12 | Ilya Rogozin | Extended payment types for Online Banking transactions with a new SPEI code. |
| 49.1 | 2025/02/24 | Artur Veletskiy | Enable Billing Transactions API for all the instances. |

# Introduction

You can get cURL from https://curl.haxx.se

This document describes the usage of the payment gateway XML/JSON/GraphQL API.

The API allows you to trigger all supported transactions of the gateway and to retrieve information about transactions existing in the gateway. You can also retrieve chargeback information.

The payment API is synchronous (except for 3-D secure payments), it accepts HTTP POST or XML data and returns XML/JSON data. Connections are always secured via SSL both in test and live mode. Be sure to set Content-type: text/xml in your headers.

> The API cannot be accessed via HTTP GET.

> Current Java SDKs do not have the necessary root certificate installed. You need to download the CA with your browser and import it into cacerts manually.

# Audience

This document is intended for technical staff integrating the XML/JSON API in the merchant’s organization.

It is required that readers have working knowledge of programming languages, XML and JSON formats and UTF8 encodings.

# Authentication

Send username and password directly in url

```
curl https://username:password@staging.gate.e-comprocessing.net:443/process/TERMINAL-TOKEN
```

Or use -u flag

```
curl -u username:password https://staging.gate.e-comprocessing.net:443/process/TERMINAL-TOKEN
```

To interact with the payment API, you need to provide login credentials using standard HTTP Basic Authentication. (credentials can be found in your Admin interface.)

To decrease network traffic and response times, we recommend that you enforce sending authentication credentials directly in the first request.

Some implementations like e.g. the Java HTTPClient automatically try to guess the best authentication scheme. This can be overridden by setting the authorization to preemptive:

> HTTPClient.getParams.setAuthenticationPreemptivetrue;

# Environments

There are two environments which can be used by merchants - **production** and **staging**, also known as **test** environment.

The **staging** environment is used only for **TESTING** purposes. When you are processing a payment using your configuration for the **test environment** the payment **WILL NOT** bring a real financial impact. This environment is used only to simulate and test different payment scenarios.

In the **production** environment you cannot use any simulated test data. When you are processing payments using your configuration for the **production environment** the payment **WILL** bring a real financial impact.

# URLs

## Server API

API methods are called with the following structure:

`https://gate.e-comprocessing.net/process/TERMINAL_TOKEN/`

Single transaction reconciles can be done via this URL:

`https://gate.e-comprocessing.net/reconcile/TERMINAL_TOKEN/`

Date range reconciles can be done via this URL:

`https://gate.e-comprocessing.net/reconcile/by_date/TERMINAL_TOKEN/`

This URL allows retrieval of a list of supported banks based on the customers country and/or currency:

`https://gate.e-comprocessing.net/retrieve_inpay_banks/TERMINAL_TOKEN/`

For the test system use the following URLs:

`https://staging.gate.e-comprocessing.net/process/TERMINAL_TOKEN/`

`https://staging.gate.e-comprocessing.net/reconcile/TERMINAL_TOKEN/`

`https://staging.gate.e-comprocessing.net/reconcile/by_date/TERMINAL_TOKEN/`

`https://staging.gate.e-comprocessing.net/retrieve_inpay_banks/TERMINAL_TOKEN/`

# WPF

The URL for the WPF API create method is:

`https://wpf.e-comprocessing.net/wpf`

For the test system the URL is:

`https://staging.wpf.e-comprocessing.net/wpf`

The URL for the WPF API reconcile method is:

`https://wpf.e-comprocessing.net/wpf/reconcile`

For the test system the URL is:

`https://staging.wpf.e-comprocessing.net/wpf/reconcile`

> Terminal token and login credentials can be found in your admin panel. To use our testing environment, refer to chapter Testing

Your admin panel can be found here:

`https://merchant.e-comprocessing.net/`

# Consumer API

Create consumer:

`https://gate.e-comprocessing.net/v1/create_consumer/`

Retrieve consumer:

`https://gate.e-comprocessing.net/v1/retrieve_consumer/`

Update consumer:

`https://gate.e-comprocessing.net/v1/update_consumer/`

Disable consumer:

`https://gate.e-comprocessing.net/v1/disable_consumer/`

Enable consumer:

`https://gate.e-comprocessing.net/v1/enable_consumer/`

Get consumer cards:

`https://gate.e-comprocessing.net/v1/get_consumer_cards/`

For the test system use the following URLs:

`https://staging.gate.e-comprocessing.net/v1/create_consumer/`

`https://staging.gate.e-comprocessing.net/v1/update_consumer/`

`https://staging.gate.e-comprocessing.net/v1/disable_consumer/`

`https://staging.gate.e-comprocessing.net/v1/enable_consumer/`

`https://staging.gate.e-comprocessing.net/v1/get_consumer_cards/`

# Tokenization API

Tokenize cardholder data:

`https://gate.e-comprocessing.net/v1/tokenize/`

Detokenize cardholder data:

`https://gate.e-comprocessing.net/v1/detokenize/`

Update the cardholder data (PAN cannot be updated):

`https://gate.e-comprocessing.net/v1/update_token/`

Validate if given token is valid for the merchant:

`https://gate.e-comprocessing.net/v1/validate_token/`

Delete a token:

`https://gate.e-comprocessing.net/v1/delete_token/`

Exchange masked cardholder data for token:

`https://gate.e-comprocessing.net/v1/get_card/`

For the test system use the following URLs:

`https://staging.gate.e-comprocessing.net/v1/tokenize/`

`https://staging.gate.e-comprocessing.net/v1/detokenize/`

`https://staging.gate.e-comprocessing.net/v1/update_token/`

`https://staging.gate.e-comprocessing.net/v1/validate_token/`

`https://staging.gate.e-comprocessing.net/v1/delete_token/`

`https://staging.gate.e-comprocessing.net/v1/get_card/`

> Terminal token and login credentials can be found in your admin panel. To use our testing environment, refer to chapter Testing

# Genesis KYC Services JSON API

**Testing environment**

Create Consumer:

`POST https://staging.kyc.e-comprocessing.net/api/v1/create_consumer`

Update Consumer:

`POST https://staging.kyc.e-comprocessing.net/api/v1/update_consumer`

Create Transaction:

`POST https://staging.kyc.e-comprocessing.net/api/v1/create_transaction`

Update Transaction:

`POST https://staging.kyc.e-comprocessing.net/api/v1/update_transaction`

Upload Document:

`POST https://staging.kyc.e-comprocessing.net/api/v1/upload_document`

Download Document:

`POST https://staging.kyc.e-comprocessing.net/api/v1/download_document`

Verify Phone:

`POST https://staging.kyc.e-comprocessing.net/api/v1/verify_phone`

Verify Identity:

`POST https://staging.kyc.e-comprocessing.net/api/v1/verify_identity`

Verify Bank Account:

`POST https://staging.kyc.e-comprocessing.net/api/v1/verify_bank_account`

Create Authentication:

`POST https://staging.kyc.e-comprocessing.net/api/v1/create_authentication`

Update Authentication:

`POST https://staging.kyc.e-comprocessing.net/api/v1/update_authentication`

Create Verification:

`POST https://staging.kyc.e-comprocessing.net/api/v1/verifications`

Check Verification Status:

`POST https://staging.kyc.e-comprocessing.net/api/v1/verifications/status`

Register Reference_ID:

`POST https://staging.kyc.e-comprocessing.net/api/v1/verifications/register`

**Production environment**

Genesis KYC Services JSON API URLs for production environment:

Create Consumer:

`POST https://kyc.e-comprocessing.net/api/v1/create_consumer`

Update Consumer:

`POST https://kyc.e-comprocessing.net/api/v1/update_consumer`

Create Transaction:

`POST https://kyc.e-comprocessing.net/api/v1/create_transaction`

Update Transaction:

`POST https://kyc.e-comprocessing.net/api/v1/update_transaction`

Upload Document:

`POST https://kyc.e-comprocessing.net/api/v1/upload_document`

Download Document:

`POST https://kyc.e-comprocessing.net/api/v1/download_document`

Verify Phone:

`POST https://kyc.e-comprocessing.net/api/v1/verify_phone`

Verify Identity:

`POST https://kyc.e-comprocessing.net/api/v1/verify_identity`

Verify Bank Account:

`POST https://kyc.e-comprocessing.net/api/v1/verify_bank_account`

Create Authentication:

`POST https://kyc.e-comprocessing.net/api/v1/create_authentication`

Update Authentication:

`POST https://kyc.e-comprocessing.net/api/v1/update_authentication`

Create Verification:

`POST https://kyc.e-comprocessing.net/api/v1/verifications`

Check Verification Status:

`POST https://kyc.e-comprocessing.net/api/v1/verifications/status`

---

# Transaction API

Update expiration date:

`PUT https://gate.e-comprocessing.net/v1/transaction/expiry_date/:transaction_unique_id`

For the test system use the following URLs:

`PUT https://staging.gate.e-comprocessing.net/v1/transaction/expiry_date/:transaction_unique_id`

---

# Smart Routing API

For the test system use the following URLs:

`POST https://staging.api.e-comprocessing.net/transactions`

With authentication params in the URL:

`POST https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.api.e-comprocessing.net/transactions`

For the production system use the following URLs:

`POST https://prod.api.e-comprocessing.net/transactions`

With authentication params in the URL:

`POST https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@prod.api.e-comprocessing.net/transactions`

---

## Billing Transactions API

**Testing environment**

`POST https://staging.api.e-comprocessing.net/billing_transactions/v1/graphql`

**Production environment**

`POST https://prod.api.e-comprocessing.net/billing_transactions/v1/graphql`

---

# Transactions

---

## Invoking a Transaction

A transaction is invoked via HTTPS POST, parameters are passed as XML with UTF-8 encoding.

---

## Card

**RECURRING V2**

A recurring transaction describes a payment where the cardholder’s account is periodically charged for a repeated delivery and use of a product or service (subscription, membership fee, etc.) over time. A recurring payment consists of an initial transaction and one or several repeated subsequent transactions. The ”initial” transaction contains all relevant card and cardholder data, while the subsequent repeated transaction references an identifier which is returned with the response to the initial request.

**INITIAL RECURRING**

Sale(3D) or Authorize(3D) with recurring_type initial marks an initialization of recurring series. Subsequent recurring transactions use initial Sale(3D) or Authorize(3D) as a reference for the recurring series.

Note that if an initial recurring is fully refunded or referenced by a fraud transaction(ChargebackTransaction / ChargebackReversalTransaction / RepresentmentTransaction / SecondChargebackTransaction), the recurring series is stopped and no more Subsequent recurring transactions can be performed for that recurring series.

If an initial recurring is partially refunded, the recurring series can continue with more Subsequent recurring transactions.

This transaction type supports Tokenization.

This transaction type supports Level 3 travel data.

This transaction type could require business attributes.

This transaction type supports Managed Recurring.

This transaction type supports Account Name Inquiry attributes.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <recurring_type>initial</recurring_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <recurring_category>subscription</recurring_category>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
```

```
  </account_owner>
 </payment_transaction>'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize</transaction_type>
   <recurring_type>initial</recurring_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <recurring_category>subscription</recurring_category>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sale, sale3d, authorize or authorize3d** |
| recurring_type | required | string(255) | Specifies recurring type of the transaction, 'initial' |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount in order not to charge the consumer with the initial recurring, but with the followed RecurringSale transactions only. For more information regarding the use cases and scenario, Contact tech-support@e-comprocessing.com for more details. |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| scheme_tokenized | required* | "true" | Required when the `card_number` is DPAN instead of Funding Primary Account Number, see Tokenized e-commerce for details |
| recurring_category | optional | string | Specifies whether the recurring transaction is a subscription(fixed amount, fixed intervals)or if it is a standing order(varying amount, fixed intervals). The allowed values are `subscription` and `standing_order`. The default value is `subscription` |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| credential_on_file_transaction_identifier | optional | string(15..32) | See Credential On File (COF) for more details |
| credential_on_file_settlement_date | optional | string(4) | See Credential On File (COF) for more details |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |

| | | | |
|---|---|---|---|
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **account_owner** | optional | | Account owner parameters related to account owner inquiry requests |
| first_name | optional | string(35) | Account owner first name |
| middle_name | optional | string(35) | Account owner middle name |
| last_name | optional | string(35) | Account owner last name |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <recurring_type>initial</recurring_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <cvv_result_code>M</cvv_result_code>
  <authorization_code>345678</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:30Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
```

```
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <scheme_response_code>00</scheme_response_code>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
  <account_owner>
    <first_name>Travis</first_name>
    <middle_name>Joe</middle_name>
    <last_name>Pastrana</last_name>
  </account_owner>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| recurring_type | string(255) | The recurring type(initial, subsequent or managed) |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ‘true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |

Error Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <recurring_type>initial</recurring_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <message>billing_address[zip_code] is invalid!</message>
  <timestamp>2025-02-26T16:01:30Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```


**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| recurring_type | string(255) | The recurring type(initial, subsequent or managed) |
| status | string(255) | Status of the transaction, see states |

| | | |
|---|---|---|
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

#### SUBSEQUENT RECURRING

A Subsequent recurring transaction is a "repeated" transaction that follows and references an Initial recurring transaction.

The card and cardholder data is omitted. Note that Subsequent recurring can be partially or fully refunded if the configuration allows it, and this will not stop the recurring series.

This transaction type supports Level 3 travel data.

Business attributes are optional, but if submitted they will override the already supplied attributes in the initial sale / sale3d / authorize / authorize3d transaction.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <recurring_type>subsequent</recurring_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
   <amount>100</amount>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
 </payment_transaction>'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize</transaction_type>
   <recurring_type>subsequent</recurring_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
   <amount>100</amount>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sale or authorize** |
| recurring_type | required | string(255) | Specifies recurring type of the transaction, 'subsequent' |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |

| | | | |
|---|---|---|---|
| delivery_date | required | dd.mm.yyyy | Date of the expected delivery in format dd.mm.yyyy |
| name_of_the_supplier | required* | string | |

required* = conditionally required

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <recurring_type>subsequent</recurring_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <authorization_code>345678</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:30Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_response_code>00</scheme_response_code>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| recurring_type | string(255) | The recurring type(initial, subsequent or managed) |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <recurring_type>subsequent</recurring_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <message>billing_address[zip_code] is invalid!</message>
  <timestamp>2025-02-26T16:01:30Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| recurring_type | string(255) | The recurring type(initial, subsequent or managed) |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |

| | | |
|---|---|---|
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**NON-REFERENCED SUBSEQUENT RECURRING**

To allow for frictionless migration of merchant recurring traffic from other gateways without requiring an initial recurring transaction yet again Sale or Authorize with recurring_type subsequent does not need to use initial Sale or Authorize as a reference for the recurring series in the cases when the `credential_on_file_transaction_identifier` and `credential_on_file_settlement_date` parameters are provided in the transaction request for specific acquirers, please contact Tech Support for more details.

This transaction type supports Tokenization.

This transaction type supports Level 3 travel data.

This transaction type could require business attributes.

This transaction type supports Account Name Inquiry attributes.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <recurring_type>subsequent</recurring_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <credential_on_file_transaction_identifier>MCSQF8SOC</credential_on_file_transaction_identifier>
   <credential_on_file_settlement_date>0107</credential_on_file_settlement_date>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <recurring_category>subscription</recurring_category>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
   </account_owner>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **authorize or sale** |
| recurring_type | required | string(255) | Specifies recurring type of the transaction, 'subsequent' |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount in order not to charge the consumer with the initial recurring, but with the followed RecurringSale transactions only. For more information regarding the use cases and scenario, Contact tech-support@e-comprocessing.com for more details. |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |

| | | | |
|---|---|---|---|
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| scheme_tokenized | required* | "true" | Required when the `card_number` is DPAN instead of Funding Primary Account Number, see Tokenized e-commerce for details |
| recurring_category | optional | string | Specifies whether the recurring transaction is a subscription(fixed amount, fixed intervals)or if it is a standing order(varying amount, fixed intervals). The allowed values are `subscription` and `standing_order`. The default value is `subscription` |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| credential_on_file_transaction_identifier | required | string(9..15) | See Credential On File (COF) for more details. Required for Non-referenced Recurring transactions. The length of the value should be 15 for Visa cards, and 9..12 for Mastercard/Maestro cards |
| credential_on_file_settlement_date | required* | | See Credential On File (COF) for more details. Required for Mastercard/Maestro cards |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **account_owner** | optional | | Account owner parameters related to account owner inquiry requests |
| first_name | optional | string(35) | Account owner first name |
| middle_name | optional | string(35) | Account owner middle name |
| last_name | optional | string(35) | Account owner last name |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |

| | | | |
|---|---|---|---|
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <recurring_type>subsequent</recurring_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <cvv_result_code>M</cvv_result_code>
  <authorization_code>345678</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:30Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <scheme_response_code>00</scheme_response_code>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
  <account_owner>
    <first_name>Travis</first_name>
    <middle_name>Joe</middle_name>
    <last_name>Pastrana</last_name>
  </account_owner>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| recurring_type | string(255) | The recurring type(initial, subsequent or managed) |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ‘true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |

| | | |
|---|---|---|
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <recurring_type>subsequent</recurring_type>
  <status>declined</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>510</code>
  <technical_message>Invalid Issuer</technical_message>
  <message>Transaction failed, please contact support!</message>
  <timestamp>2025-02-26T16:01:30Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| recurring_type | string(255) | The recurring type(initial, subsequent or managed) |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <recurring_type>subsequent</recurring_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>510</code>
  <message>Transaction failed, please contact support!</message>
  <timestamp>2025-02-26T16:01:30Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| recurring_type | string(255) | The recurring type(initial, subsequent or managed) |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

| | | |
|---|---|---|
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

#### RECURRING FOR INDIAN CARDS

Recurring transactions with cards issued in India are subject to special rules. Prior to requesting a recurring series, the merchant should register the recurring agreement as per the Reserve Bank of India (RBI) regulations. After that, use the `managed_recurring` params section in order to provide the params from the agreement. Should be sent in both Initial and Subsequent recurring transactions.

> Currently availabe for Visa only. Master and Intl Maestro will be added in the future.

#### HOW TO USE MANAGED RECURRING FOR INDIAN CARDS IN PROCESSING API

##### REQUESTS

Managed Recurring

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <recurring_type>managed</recurring_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
   <card_holder>Travis Pastrana</card_holder>
   <cvv>834</cvv>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <managed_recurring>
     <mode>manual</mode>
     <payment_type>subsequent</payment_type>
     <amount_type>fixed</amount_type>
     <frequency>weekly</frequency>
     <registration_reference_number>123434</registration_reference_number>
     <max_amount>200</max_amount>
     <max_count>99</max_count>
     <validated>true</validated>
   </managed_recurring>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| recurring_type | required | string(255) | Specifies recurring type of the transaction, 'managed' |
| **managed_recurring** | required | | |
| mode | required | String | Fill in with '**manual**'. This indicates that the merchant will manually manage the subsequent recurring transactions. |
| payment_type | required | String | Type of the current recurring transaction. Values: initial, subsequent, modification, cancellation |
| amount_type | required | String | Type of the amount. Values: fixed, max |
| frequency | required | String | Frequency of the subsequent transactions. Values: daily, twice_weekly, weekly, ten_days, fortnightly, monthly, every_two_months, trimester, quarterly, twice_yearly, annually, unscheduled |
| registration_reference_number | required | String(35) | Reference number as per the agreement. |
| max_amount | required | Integer | Maximum amount as per the agreement. |
| max_count | required | Integer | Maximum transactions count as per the agreement. 99 - indicates infinite subsequent payments. |
| validated | required | String | Indicates if the current transaction is valid as per the registered agreement. Values: true, false |

`required* = conditionally required`

#### HOW TO USE MANAGED RECURRING FOR INDIAN CARDS IN WPF API

##### REQUESTS

Merchants can send managed recurring params in the request when creating Initial recurring transactions via our WPF API.

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <wpf_payment>
   <transaction_type>wpf_create</transaction_type>
   <recurring_type>initial</recurring_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <description>You are about to buy 3 shoes at www.shoes.com!</description>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <lifetime>60</lifetime>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
   <transaction_types>
     <transaction_type>
       <name>sale</name>
```

```
      <recurring_type>managed</recurring_type>
      <managed_recurring>
        <mode>manual</mode>
        <payment_type>subsequent</payment_type>
        <amount_type>fixed</amount_type>
        <frequency>weekly</frequency>
        <registration_reference_number>123434</registration_reference_number>
        <max_amount>200</max_amount>
        <max_count>99</max_count>
        <validated>true</validated>
      </managed_recurring>
    </transaction_type>
  </transaction_types>
  <remember_card>true</remember_card>
  <wpf_payment>managed_recurring</wpf_payment>
 </wpf_payment>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| recurring_type | required | string(255) | Specifies recurring type of the transaction, 'managed' |
| **managed_recurring** | required | | |
| mode | required | String | Fill in with '**manual**'. This indicates that the merchant will manually manage the subsequent recurring transactions. |
| payment_type | required | String | Type of the current recurring transaction. Values: initial, subsequent, modification, cancellation |
| amount_type | required | String | Type of the amount. Values: fixed, max |
| frequency | required | String | Frequency of the subsequent transactions. Values: daily, twice_weekly, weekly, ten_days, fortnightly, monthly, every_two_months, trimester, quarterly, twice_yearly, annually, unscheduled |
| registration_reference_number | required | String(35) | Reference number as per the agreement. |
| max_amount | required | Integer | Maximum amount as per the agreement. |
| max_count | required | Integer | Maximum transactions count as per the agreement. 99 - indicates infinite subsequent payments. |
| validated | required | String | Indicates if the current transaction is valid as per the registered agreement. Values: true, false |

`required* = conditionally required`

**RECURRING TRANSACTIONS**

A recurring transaction describes a payment where the cardholder’s account is periodically charged for a repeated delivery and use of a product or service (subscription, membership fee, etc.) over time. A recurring payment consists of an initial transaction and one or several repeated transactions. The ”initial” transaction contains all relevant card and cardholder data, while the subsequent repeated transaction references an identifier which is returned with the response to the initial request.

**INIT RECURRING SALE**

Init Recurring Sale transanction will be soon deprecated. Please start using Sale or Authorize transaction with initial recurring type instead.

An InitRecurringSale transaction initializes a recurring payment and is equal to a normal SaleTransaction except that it can be referenced as ”initial” transaction in a RecurringSale transaction.

Note that if an InitRecurringSale is fully refunded, the recurring series is stopped and no more RecurringSales can be performed for that recurring series.

If an InitRecurringSale is partially refunded, the recurring series can continue with more RecurringSales

This transaction type supports Tokenization.

This transaction type supports Level 3 travel data.

This transaction type could require business attributes.

This transaction type supports Managed Recurring.

This transaction type supports Account Name Inquiry attributes.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>init_recurring_sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <recurring_category>subscription</recurring_category>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
```

```
    <last_name>Pastrana</last_name>
  </account_owner>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **init_recurring_sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount in order not to charge the consumer with the initial recurring, but with the followed RecurringSale transactions only. For more information regarding the use cases and scenario, Contact tech-support@e-comprocessing.com for more details. |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| scheme_tokenized | required* | "true" | Required when the `card_number` is DPAN instead of Funding Primary Account Number, see Tokenized e-commerce for details |
| recurring_category | optional | string | Specifies whether the recurring transaction is a subscription(fixed amount, fixed intervals)or if it is a standing order(varying amount, fixed intervals). The allowed values are `subscription` and `standing_order`. The default value is `subscription` |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| credential_on_file_transaction_identifier | optional | string(15..32) | See Credential On File (COF) for more details |
| credential_on_file_settlement_date | optional | string(4) | See Credential On File (COF) for more details |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |

| | | | |
|---|---|---|---|
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |
| **sender** | optional | | Funding Transaction Sender details |
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |
| **account_owner** | optional | | Account owner parameters related to account owner inquiry requests |
| first_name | optional | string(35) | Account owner first name |
| middle_name | optional | string(35) | Account owner middle name |
| last_name | optional | string(35) | Account owner last name |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>init_recurring_sale</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <cvv_result_code>M</cvv_result_code>
  <authorization_code>345678</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
```

```
  <scheme_response_code>00</scheme_response_code>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
  <account_owner>
    <first_name>Travis</first_name>
    <middle_name>Joe</middle_name>
    <last_name>Pastrana</last_name>
  </account_owner>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ‘true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>init_recurring_sale</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <message>billing_address[zip_code] is invalid!</message>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

| | | |
|---|---|---|
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

#### RECURRING SALE

Recurring Sale transanction will be soon deprecated. Please start using Sale transaction with subsequent recurring type instead.

A RecurringSale transaction is a "repeated" transaction which follows and references a Init Recurring Sale transaction.

The card and cardholder data is omitted. Note that RecurringSales can be partially or fully refunded if configuration allows it, and this will not stop the recurring series.

This transaction type supports Level 3 travel data.

Business attributes are optional, but if submitted they will override the already supplied attributes in the initial init_recurring_sale / init_recurring_sale3d transaction.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>recurring_sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
   <amount>100</amount>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **recurring_sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |

| | | | |
|---|---|---|---|
| **sender** | optional | | Funding Transaction Sender details |
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>recurring_sale</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <authorization_code>345678</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_response_code>00</scheme_response_code>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Optional, if received in the response from the issuer |
| recurring_advice_text | string | Optional, describes the specific recurring advice code |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>recurring_sale</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <message>billing_address[zip_code] is invalid!</message>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |

| | | |
|---|---|---|
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

#### RECURRING FOR INDIAN CARDS

Recurring transactions with cards issued in India are subject to special rules. Prior to requesting the recurring transaction, the merchant should register the recurring agreement as per the Reserve bank of India (RBI) regulations. After that, use the `managed_recurring` params section in order to provide the params from the agreement. Should be sent in both initial and subsequent recurring transactions.

> Currently availabe for Visa only. Master and Intl Maestro will be added in the future.

#### HOW TO USE MANAGED RECURRING FOR INDIAN CARDS IN PROCESSING API

##### REQUESTS

Managed Recurring

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
   <card_holder>Travis Pastrana</card_holder>
   <cvv>834</cvv>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <managed_recurring>
     <mode>manual</mode>
     <payment_type>subsequent</payment_type>
     <amount_type>fixed</amount_type>
     <frequency>weekly</frequency>
     <registration_reference_number>123434</registration_reference_number>
     <max_amount>200</max_amount>
     <max_count>99</max_count>
     <validated>true</validated>
   </managed_recurring>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **managed_recurring** | required | | |
| mode | required | String | Fill in with '**manual**'. This indicates that the merchant will manually manage the subsequent recurring transactions. |
| payment_type | required | String | Type of the current recurring transaction. Values: initial, subsequent, modification, cancellation |
| amount_type | required | String | Type of the amount. Values: fixed, max |
| frequency | required | String | Frequency of the subsequent transactions. Values: daily, twice_weekly, weekly, ten_days, fortnightly, monthly, every_two_months, trimester, quarterly, twice_yearly, annually, unscheduled |
| registration_reference_number | required | String(35) | Reference number as per the agreement. |
| max_amount | required | Integer | Maximum amount as per the agreement. |
| max_count | required | Integer | Maximum transactions count as per the agreement. 99 - indicates infinite subsequent payments. |
| validated | required | String | Indicates if the current transaction is valid as per the registered agreement. Values: true, false |

`required* = conditionally required`

#### HOW TO USE MANAGED RECURRING FOR INDIAN CARDS IN WPF API

##### REQUESTS

Merchants can send managed recurring params in the request when creating initial recurring transactions via our WPF API.

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <wpf_payment>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <description>You are about to buy 3 shoes at www.shoes.com!</description>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <lifetime>60</lifetime>
   <billing_address>
```

```
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
  <transaction_types>
    <transaction_type>
      <name>init_recurring_sale</name>
      <managed_recurring>
        <mode>manual</mode>
        <payment_type>subsequent</payment_type>
        <amount_type>fixed</amount_type>
        <frequency>weekly</frequency>
        <registration_reference_number>123434</registration_reference_number>
        <max_amount>200</max_amount>
        <max_count>99</max_count>
        <validated>true</validated>
      </managed_recurring>
    </transaction_type>
  </transaction_types>
  <remember_card>true</remember_card>
  <wpf_payment>managed_recurring</wpf_payment>
 </wpf_payment>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **managed_recurring** | required | | |
| mode | required | String | Fill in with '**manual**'. This indicates that the merchant will manually manage the subsequent recurring transactions. |
| payment_type | required | String | Type of the current recurring transaction. Values: initial, subsequent, modification, cancellation |
| amount_type | required | String | Type of the amount. Values: fixed, max |
| frequency | required | String | Frequency of the subsequent transactions. Values: daily, twice_weekly, weekly, ten_days, fortnightly, monthly, every_two_months, trimester, quarterly, twice_yearly, annually, unscheduled |
| registration_reference_number | required | String(35) | Reference number as per the agreement. |
| max_amount | required | Integer | Maximum amount as per the agreement. |
| max_count | required | Integer | Maximum transactions count as per the agreement. 99 - indicates infinite subsequent payments. |
| validated | required | String | Indicates if the current transaction is valid as per the registered agreement. Values: true, false |

`required* = conditionally required`

**AUTHORIZE**

With authorize transactions, you can confirm that a credit card is valid and reserve the desired amount on the card.

After settling the transaction (e.g. shipping the goods), you can then capture the amount. The customer will not be billed until the capture has taken place, but the amount is reserved and the customer's credit card limit is reduced. Authorizes will automatically be cancelled after a certain timeframe, most likely one week.

For a typical e-commerce application it is recommended to authorize the amount on incoming orders and capture it when shipping the goods. If you are selling services or non-tangible goods, you can use the sale transaction, which combines authorize and capture.

If you choose not to serve the customer, consider to void the authorize to unfreeze the amount on the client's credit card.

Authorize transactions are also available as 3dsecure transactions

Transactions of this type support **auth_network_outage** exemption.

It informs the issuer that Strong Customer Authentication (SCA) was not possible because of a major outage of the authentication network and infrastructure

This transaction type supports Tokenization.

This transaction type supports Level 3 travel data.

This transaction type supports Preauthorizations.

This transaction type could require business attributes.

This transaction type supports Account Name Inquiry attributes.

This transaction type supports Installments.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
```

```
    <customer_phone>+1987987987987</customer_phone>
    <business_attributes>
      <event_start_date>27-03-2025</event_start_date>
      <event_end_date>08-04-2025</event_end_date>
      <event_organizer_id>20192375</event_organizer_id>
      <event_id>1912</event_id>
    </business_attributes>
    <billing_address>
      <first_name>Travis</first_name>
      <last_name>Pastrana</last_name>
      <address1>Muster Str. 12</address1>
      <zip_code>10178</zip_code>
      <city>Los Angeles</city>
      <neighborhood>Hollywood</neighborhood>
      <state>CA</state>
      <country>US</country>
    </billing_address>
    <account_owner>
      <first_name>Travis</first_name>
      <middle_name>Joe</middle_name>
      <last_name>Pastrana</last_name>
    </account_owner>
  </payment_transaction>'
```

Funding Transaction Example

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <payment_transaction>transaction_type</payment_transaction>
   <transaction_id>1196432505475 01c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
   </account_owner>
   <funding>
     <identifier_type>business_disbursement</identifier_type>
     <receiver>
       <first_name>Hamza</first_name>
       <last_name>Arshad</last_name>
       <country>AF</country>
       <account_number>090078601</account_number>
       <account_number_type>iban</account_number_type>
       <address>Gulberg 21 street</address>
       <state>CN</state>
       <city>Lahore</city>
     </receiver>
   </funding>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **authorize** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| crypto | optional | "true" | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |
| preauthorization | optional | "true" | Signifies whether a preauthorization transaction is performed. Check the Preauthorizations section or contact tech support for more details. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount to act as an account verification transaction - Contact tech-support@e-comprocessing.com for more details regarding this scenario. |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |

| | | | |
|---|---|---|---|
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| scheme_tokenized | required* | "true" | Required when the `card_number` is DPAN instead of Funding Primary Account Number, see Tokenized e-commerce for details |
| recurring_type | optional | string(255) | Specifies recurring type of the transaction, can be 'initial', 'managed' or 'subsequent'. |
| installment_plan_id | optional | string | The ID of the chosen installment plan. Check Installments for more details. |
| installment_plan_reference | optional | string | The installment plan reference identifier returned by the schemes. |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| credential_on_file_transaction_identifier | optional | string(15..32) | See Credential On File (COF) for more details |
| credential_on_file_settlement_date | optional | string(4) | See Credential On File (COF) for more details |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **sca_params** | optional | | SCA params |
| exemption | optional | string | The exemption that the transaction should take advantage of. Note that the requested exemption may not be accepted due to internal risk validations. Check SCA exemption values. |
| visa_merchant_id | required[5] | string(8) | VMID assigned by Visa if participating in Trusted merchant program. |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |

| | | | |
|---|---|---|---|
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |
| **sender** | optional | | Funding Transaction Sender details |
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |
| **account_owner** | optional | | Account owner parameters related to account owner inquiry requests |
| first_name | optional | string(35) | Account owner first name |
| middle_name | optional | string(35) | Account owner middle name |
| last_name | optional | string(35) | Account owner last name |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <cvv_result_code>M</cvv_result_code>
  <authorization_code>345678</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_pan_indicator>V</scheme_pan_indicator>
  <payment_response>scheme_pan_tail</payment_response>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <scheme_response_code>00</scheme_response_code>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
  <account_owner>
    <first_name>Travis</first_name>
    <middle_name>Joe</middle_name>
    <last_name>Pastrana</last_name>
  </account_owner>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |

| | | |
|---|---|---|
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| payment_account_reference | string(255) | Payment Account Reference value returned from the schemes. Links tokenized and PAN-based transactions. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ‘true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_transaction_link_id | string(22) | The transaction unique identifier returned from the schemes. Corresponds to ADDITIONAL SERVICE DATA (field 65, MasterCard Transaction Link Id [TLID]). |
| scheme_pan_indicator | string(1) | The account number indicator in scheme-tokenized transactions. See Account number indicator codes for details. |
| scheme_pan_tail | string(4) | The tail of the PAN number tail associated with the token in scheme-tokenized transactions. Returned in conjunction with `scheme_pan_indicator`. |
| scheme_max_settlement_date | string(10) | The final date an authorization could be captured which is returned from the card scheme. See Scheme Maximum Settlement Date for more details. |
| scheme_authentication_data_quality | boolean | The Authentication Data Quality Indicator returned by the card scheme is used to indicate whether a transaction meets the authentication data quality requirements. The value can be either `true` or `false`. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Additional response code returned from the schemes. See Recurring advice details |
| recurring_advice_text | string(255) | The text representation of the recurring advice code. |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>expiration_year is invalid</technical_message>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |

| | | |
|---|---|---|
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**CAPTURE**

Capture settles a transaction which has been authorized before.

Do this when you are shipping goods, for example. A capture can only be used after an authorize on the same transaction and on the same terminal.

Therefore, the reference id of the authorized transaction is mandatory.

You can also capture a partial amount of the initially authorized amount, e.g. if you want to give customers a discount. However, you cannot capture a higher amount than initially authorized.

This transaction type supports Level 3 travel data.

This transaction can be used to capture a Preauthorization.

This transaction type supports Installments.

Business attributes are optional, but if submitted they will override the already supplied attributes in the initial authorize / authorize3d transaction.

Transaction workflow:

1. The merchant sends authorize transaction to the gateway.
2. The gateway replies to it. One of returned values is the unique id of the transaction.
3. The merchant sends capture transaction. Its reference id is unique id of authorize response.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>capture</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
   <amount>100</amount>
   <currency>USD</currency>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **capture** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>capture</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <authorization_code>345678</authorization_code>
   <response_code>00</response_code>
   <technical_message>Transaction successful!</technical_message>
   <message>Transaction successful!</message>
   <timestamp>2025-02-26T16:01:31Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
```

```
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>capture</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>430</code>
  <technical_message>Reference transaction has already been captured, and acquirer does not support partial/multiple capture</technical_message>
  <message>Transaction declined.</message>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**SALE**

Sale transactions combine authorize and capture into one step.

Using a sale transaction, the amount is immediately billed to the customer’s credit card. It can be reversed via a void transaction on the same day of the transaction. Use sale transactions, if you are e.g. selling non-tangible goods or services.

Sale transactions are also available as 3dsecure transactions

Transactions of this type support **auth_network_outage** exemption.

It informs the issuer that Strong Customer Authentication (SCA) was not possible because of a major outage of the authentication network and infrastructure

This transaction type supports Tokenization.

This transaction type supports Level 3 travel data.

This transaction type could require business attributes.

This transaction type supports Account Name Inquiry attributes.

This transaction type supports Installments.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
   </account_owner>
 </payment_transaction>'
```

Funding Transaction Example

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <payment_transaction>transaction_type</payment_transaction>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
   </account_owner>
   <funding>
     <identifier_type>business_disbursement</identifier_type>
     <receiver>
       <first_name>Hamza</first_name>
       <last_name>Arshad</last_name>
       <country>AF</country>
       <account_number>090078601</account_number>
       <account_number_type>iban</account_number_type>
       <address>Gulberg 21 street</address>
       <state>CN</state>
       <city>Lahore</city>
     </receiver>
   </funding>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| crypto | optional | "true" | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |

| | | | |
|---|---|---|---|
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount to act as an account verification transaction - Contact tech-support@e-comprocessing.com for more details regarding this scenario. |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| scheme_tokenized | required* | "true" | Required when the `card_number` is DPAN instead of Funding Primary Account Number, see Tokenized e-commerce for details |
| recurring_type | optional | string(255) | Specifies recurring type of the transaction, can be 'initial', 'managed' or 'subsequent'. |
| reference_id | optional | string(32) | Unique id returned by corresponding transaction |
| installment_plan_id | optional | string | The ID of the chosen installment plan. Check Installments for more details. |
| installment_plan_reference | optional | string | The installment plan reference identifier returned by the schemes. |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| credential_on_file_transaction_identifier | optional | string(15..32) | See Credential On File (COF) for more details |
| credential_on_file_settlement_date | optional | string(4) | See Credential On File (COF) for more details |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **sca_params** | optional | | SCA params |

| | | | |
|---|---|---|---|
| exemption | optional | string | The exemption that the transaction should take advantage of. Note that the requested exemption may not be accepted due to internal risk validations. Check SCA exemption values. |
| visa_merchant_id | required[5] | string(8) | VMID assigned by Visa if participating in Trusted merchant program. |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |
| **sender** | optional | | Funding Transaction Sender details |
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |
| **account_owner** | optional | | Account owner parameters related to account owner inquiry requests |
| first_name | optional | string(35) | Account owner first name |
| middle_name | optional | string(35) | Account owner middle name |
| last_name | optional | string(35) | Account owner last name |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <cvv_result_code>M</cvv_result_code>
  <authorization_code>345678</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_pan_indicator>V</scheme_pan_indicator>
  <payment_response>scheme_pan_tail</payment_response>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <scheme_response_code>00</scheme_response_code>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
```

```
  <sca_exemption_result>13</sca_exemption_result>
  <account_owner>
    <first_name>Travis</first_name>
    <middle_name>Joe</middle_name>
    <last_name>Pastrana</last_name>
  </account_owner>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| payment_account_reference | string(255) | Payment Account Reference value returned from the schemes. Links tokenized and PAN-based transactions. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to 'true' if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_transaction_link_id | string(22) | The transaction unique identifier returned from the schemes. Corresponds to ADDITIONAL SERVICE DATA (field 65, MasterCard Transaction Link Id [TLID]). |
| scheme_pan_indicator | string(1) | The account number indicator in scheme-tokenized transactions. See Account number indicator codes for details. |
| scheme_pan_tail | string(4) | The tail of the PAN number tail associated with the token in scheme-tokenized transactions. Returned in conjunction with `scheme_pan_indicator`. |
| scheme_max_settlement_date | string(10) | The final date an authorization could be captured which is returned from the card scheme. See Scheme Maximum Settlement Date for more details. |
| scheme_authentication_data_quality | boolean | The Authentication Data Quality Indicator returned by the card scheme is used to indicate whether a transaction meets the authentication data quality requirements. The value can be either `true` or `false`. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Additional response code returned from the schemes. See Recurring advice details |
| recurring_advice_text | string(255) | The text representation of the recurring advice code. |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <technical_message>billing_address[zip_code] is invalid!</technical_message>
  <message>billing_address[zip_code] is invalid!</message>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |

| | | |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**ARGENCARD**

Argencard is a debit or credit card used in Argentina. It allows online shoppers to pay offline for their online purchases at over 150,000 physical outlets.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>argencard</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>barney@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Buenos Aires</city>
     <neighborhood>Recoleta</neighborhood>
     <country>AR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **argencard** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National Identifier number of the customer |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |

| | | | |
|---|---|---|---|
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| AR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>argencard</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>argencard</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:31Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

| | | |
|---|---|---|
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**AURA**

Aura is a local Brazilian credit card.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>aura</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>barney@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Salvador</city>
     <neighborhood>Historic Center of Salvador</neighborhood>
     <country>BR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **aura** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National Identifier number of the customer |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |

| | | | |
|---|---|---|---|
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| BR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>aura</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>aura</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**BANCONTACT**

Bancontact is a local Belgian debit card scheme. All Belgian debit cards are co-branded Bancontact and Maestro.

Transaction flow for a consumer is identical to a Maestro payment.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>bcmc</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>EUR</currency>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>12114</zip_code>
     <city>Brussels</city>
     <neighborhood>City Center</neighborhood>
     <country>BE</country>
   </billing_address>
 </payment_transaction>'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>ppro</transaction_type>
   <payment_type>bcmc</payment_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>EUR</currency>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>12114</zip_code>
     <city>Brussels</city>
     <neighborhood>City Center</neighborhood>
     <country>BE</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | **ppro** or **bcmc**. Contact tech support at tech-support@e-comprocessing.com for more details. |
| payment_type | required* | bcmc | Bancontact Mr. Cash. Contact tech support for more details |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |

| | | | |
|---|---|---|---|
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Supported currencies and countries:

| Currency code | Country code |
|---|---|
| EUR | BE |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bcmc</transaction_type>
  <status>pending_async</status>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ppro</transaction_type>
  <status>pending_async</status>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bcmc</transaction_type>
  <status>error</status>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ppro</transaction_type>
  <status>error</status>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <code>110</code>
  <payment_response>technical_message</payment_response>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**CABAL**

Cabal is a local debit/credit card brand in Argentina which can be used for online purchases.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>cabal</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>braney_rubble</consumer_reference>
   <national_id>8812128812</national_id>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **cabal** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National Identifier number of the customer |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

| | | | |
|---|---|---|---|
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries**

| Country Name | Country Code |
|---|---|
| Argentina | AR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <status>pending_async</status>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61d0</redirect_url>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <status>error</status>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**CENCOSUD**

Cencosud is a local credit card in Argentina

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>cencosud</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>barney@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Buenos Aires</city>
     <neighborhood>Recoleta</neighborhood>
     <country>AR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **cencosud** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National Identifier number of the customer |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| AR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>cencosud</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>cencosud</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### CREDIT (CFT)

Credits (also known as Credit Fund Transfer a.k.a. CFT) can be done with an initial reference transaction.

This transaction type allows you to transfer funds to a previously charged card. The amount can be higher than the charged reference. Credits can only be done on former sale, sale3d, init recurring sale, init recurring sale3d, recurring sale or capture transaction. Therefore, the **reference_id** for the corresponding transaction is mandatory.

Both Visa and Mastercard/Maestro credits are authorized real-time.

Note that for exceptional cases with some countries Visa OCTS will not be authorized through the schemes but batched for offline settlement on the same day. This means that the authorization code and issuer response code will not be available only for them.

Note that VISA OCT transactions with Australian or Canadian card bins will require the merchant zip code to be set, either through the dynamic descriptor parameter or through the merchant configuration.

> This transaction type supports Account Name Inquiry attributes.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>credit</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <reference_id>43672</reference_id>
   <amount>100</amount>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
   </account_owner>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **credit** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| crypto | optional | "true" | Signifies whether a crypto-currency transaction is performed. Must be populated when indicating crypto for VISA and MCC 6051. This is only applied to VISA OCT transactions. Contact Tech Support for more details. |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| source_of_funds | optional | string | Specify the source of funds with one of `credit`, `debit`, `prepaid`, `cash`, `other_debit_account`, `other_credit_account`. |
| purpose_of_payment | required* | string (12) | Purpose of Payment code, required for Visa OCTs with recipients in Argentina, Bangladesh, Egypt, India, Chile and Colombia. |
| **customer_identification** | required* | | See Customer Identification Parameters for more details. |
| owner | required* | string(255) | The owner of the document ID |
| type | required* | string(255) | The type of the document ID |
| subtype | required* | string(255) | The subtype of the document ID |
| document_id | required* | string(255) | Document ID value. |
| issuing_country | required* | string(2) | The issuing country of the document ID |
| **account_owner** | optional | | Account owner parameters related to account owner inquiry requests |
| first_name | optional | string(35) | Account owner first name |
| middle_name | optional | string(35) | Account owner middle name |
| last_name | optional | string(35) | Account owner last name |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>credit</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <authorization_code>345678</authorization_code>
   <retrieval_reference_number>016813015184</retrieval_reference_number>
   <response_code>00</response_code>
   <timestamp>2025-02-26T16:01:32Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
   <scheme_response_code>00</scheme_response_code>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
   </account_owner>
 </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |

| | | |
|---|---|---|
| recurring_advice_code | string(2) | Additional response code returned from the schemes. See Recurring advice details |
| recurring_advice_text | string(255) | The text representation of the recurring advice code. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>credit</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>410</code>
  <technical_message>No approved reference transaction found</technical_message>
  <message>No approved reference transaction found</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

**ELO**

Elo is a local Brazilian payment card.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>elo</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>barney@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Salvador</city>
     <neighborhood>Historic Center of Salvador</neighborhood>
     <country>BR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **elo** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National Identifier number of the customer |

| | | | |
|---|---|---|---|
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| BR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>elo</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>elo</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
```

```
    <currency>USD</currency>
    <sent_to_acquirer>true</sent_to_acquirer>
  </payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**NARANJA**

Naranja is a local credit card issued in Argentina which can be used for purchases over the internet.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>naranja</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>barney@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Buenos Aires</city>
     <country>AR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **naranja** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National Identifier number of the customer |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |

| | | | |
|---|---|---|---|
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| AR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>naranja</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>naranja</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

| | | |
|---|---|---|
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**NATIVA**

Nativa is an Argentinian credit card provided by the National Bank of Argentina.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>nativa</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>barney@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Buenos Aires</city>
     <country>AR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **nativa** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National Identifier number of the customer |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |

| | | | |
|---|---|---|---|
| country | optional | string(2) | Country code in ISO 3166 |

required* = conditionally required

**Supported countries:**

| Country |
|---|
| AR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>nativa</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>nativa</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:32Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### TARJETA SHOPPING

Tarjeta Shopping is a cash payment in Argentina.

Warning: We do not recommend using IFrames. This causes the scheme’s pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>tarjeta_shopping</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rubble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Rosario</city>
     <country>AR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **tarjeta_shopping** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National Identifier number of the customer |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| AR |

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>tarjeta_shopping</transaction_type>
   <status>pending_async</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
```

```
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:33Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>tarjeta_shopping</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:33Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

---

# Non-financial Transactions

#### ACCOUNT VERIFICATION V2

Account Verification transaction can be used to verify the account's existence for a given cardholder without any financial impact. To create an Account Verification transaction, you have to submit a Sale, Sale(3D), Authorize or Authorize(3d) transaction with **zero amount**.

If the transaction is submitted with amount > 0 it will result in a transaction with financial impact.

Account Verification transaction can also carry on an AVS request thus you can also get the AVS response code and text from the schemes. Refer to the section AVS Status Codes for more information.

Example Of Sale(3d) Transaction As Account Verification

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
```

```
<payment_transaction>
  <transaction_type>sale3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>0</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4200000000000000</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <business_attributes>
    <event_start_date>27-03-2025</event_start_date>
    <event_end_date>08-04-2025</event_end_date>
    <event_organizer_id>20192375</event_organizer_id>
    <event_id>1912</event_id>
  </business_attributes>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

Example Of Authorize(3d) Transaction As Account Verification

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>authorize3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>0</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4200000000000000</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <business_attributes>
    <event_start_date>27-03-2025</event_start_date>
    <event_end_date>08-04-2025</event_end_date>
    <event_organizer_id>20192375</event_organizer_id>
    <event_id>1912</event_id>
  </business_attributes>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sale, sale(3d), authorize or authorize(3d)** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| crypto | optional | "true" | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer = 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. By submitting zero value amount you can verify the account's existence for a given cardholder without any financial impact. |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| scheme_tokenized | required* | "true" | Required when the `card_number` is DPAN instead of Funding Primary Account Number, see Tokenized e-commerce for details |
| recurring_type | optional | string(255) | Specifies recurring type of the transaction, can be 'initial', 'managed' or 'subsequent'. |
| reference_id | optional | string(32) | Unique id returned by corresponding transaction |

| | | | |
|---|---|---|---|
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| credential_on_file_transaction_identifier | optional | string(15..32) | See Credential On File (COF) for more details |
| credential_on_file_settlement_date | optional | string(4) | See Credential On File (COF) for more details |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **sca_params** | optional | | SCA params |
| exemption | optional | string | The exemption that the transaction should take advantage of. Note that the requested exemption may not be accepted due to internal risk validations. Check SCA exemption values. |
| visa_merchant_id | required[5] | string(8) | VMID assigned by Visa if participating in Trusted merchant program. |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

required* = conditionally required

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <cvv_result_code>P</cvv_result_code>
  <authorization_code>271621</authorization_code>
  <retrieval_reference_number>311709000149</retrieval_reference_number>
  <response_code>83</response_code>
  <timestamp>2025-02-26T16:01:33Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>0</amount>
  <currency>USD</currency>
  <scheme_transaction_identifier>427105912289261</scheme_transaction_identifier>
  <scheme_response_code>00</scheme_response_code>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ‘true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Additional response code returned from the schemes. See Recurring advice details |
| recurring_advice_text | string(255) | The text representation of the recurring advice code. |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>billing_address[zip_code] is invalid!</technical_message>
  <message>billing_address[zip_code] is invalid!</message>
  <timestamp>2025-02-26T16:01:33Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>0</amount>
  <currency>USD</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |

| | | |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**ACCOUNT VERIFICATION**

Account Verification transactions are implemented using the so-called zero-value auths.

Using an account verification transaction, the account existence for a given cardholder can be verified without any financial impact.

Note the account verification can also carry on an AVS request, thus you can also get the AVS response code and text by the schemes along with it. Refer to section AVS Status Codes for more information.

This transaction type supports Tokenization.

Account verification transaction will be soon deprecated. You can use Authorize, Authorize 3D, Sale or Sale 3D as account verification by processing transaction with **zero amount**.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>account_verification</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

Example When Issuer Supports Oct For This Pan:

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>account_verification</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <issuer_oct_enabled>true</issuer_oct_enabled>
   <remote_ip>245.253.2.12</remote_ip>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **account_verification** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |

| | | | |
|---|---|---|---|
| usage | optional | string(255) | Description of the transaction for later use. |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| issuer_oct_enabled | optional | true | Supported only by Visa. When submitted, Visa checks if the given PAN supports OCTs at the issuer. When not submitted, it is interpreted as a normal account verification. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| document_id | required* | string(255) | Document ID value. |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>account_verification</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:33Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <scheme_response_code>00</scheme_response_code>
</payment_response>
```

Example When Issuer Supports Oct For This Pan:

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>account_verification</transaction_type>
  <status>approved</status>
  <issuer_oct_enabled>true</issuer_oct_enabled>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
```

```
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:33Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <scheme_response_code>00</scheme_response_code>
</payment_response>
```


Example When Issuer Does Not Support Oct For This Pan:

Successful Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>account_verification</transaction_type>
  <status>declined</status>
  <issuer_oct_enabled>false</issuer_oct_enabled>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:33Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <scheme_response_code>00</scheme_response_code>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| issuer_oct_enabled | string | Present only if merchant has submitted issuer oct enabled flag in the request to check if issuer supports OCTs for the given PAN. True if the issuer supports OCTs for this PAN, false otherwise. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| sent_to_acquirer | string(255) | "true" or "false" |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Additional response code returned from the schemes. See Recurring advice details |
| recurring_advice_text | string(255) | The text representation of the recurring advice code. |

Error Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>account_verification</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>expiration_year is invalid</technical_message>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:33Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

---

# 3DS Card

To learn more about 3D Secure and supported authentication protocols, please, visit 3D-Secure Section.

**AUTHORIZE 3D**

Authorize3D transactions basically have the same request as standard authorize transactions.

Authorize3D transactions can be handled synchronous or asynchronous depending on the parameters passed. If mpi params is passed the workflow will be synchronous. If notification url, return success url and return failure url are passed the workflow will be asynchronous.

To settle Authorize3D transactions, normal capture transactions are used. As the 3-D secure process already took place, there is no need to do the verification again when capturing.

This transaction type supports Tokenization.

This transaction type supports Level 3 travel data.

This transaction type supports Partial Approvals.

This transaction type supports Preauthorizations.

This transaction type could require business attributes.

An exemption from Strong Customer Authentication (SCA) can be requested by submitting an **exemption** with `low_risk` under SCA params.

In case the issuer accepts the exemption, a step up in the authentication flow might not be required because the transaction's risk analysis has already been performed by acquirer.

Note, the requested exemption might not be accepted due to internal risk validations.

For example, to be able to utilize the low risk exemption, the BIN country of the card must be part of the European Economic Area (EEA).

Furthermore, the acquirer could accept the merchant low-risk exemption request only if the transaction amount does not exceed the acquirer low-risk exemption threshold.

Finally, the ACS might not acknowledge the merchant/acquirer's exemption request and may still require a step up in the cardholder authentication.

This transaction type supports Installments.

Visa Synchronous 3 D Sv2 Fully Authenticated Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4012000000060085</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
```

```
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <mpi_params>
    <eci>05</eci>
    <cavv>MDAwMDAwMDAwMDAxMTA2Njk5NFg=</cavv>
    <protocol_version>2</protocol_version>
    <protocol_sub_version>2</protocol_sub_version>
    <directory_server_id>09fbdb30-d752-013d-a6e2-0a58a9feac02</directory_server_id>
    <acs_transaction_id>09fbdc70-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
    <threeds_challenge_indicator>preference</threeds_challenge_indicator>
  </mpi_params>
  <sca_params>
    <exemption>trusted_merchant</exemption>
    <visa_merchant_id>00000000</visa_merchant_id>
  </sca_params>
 </payment_transaction>'
```

Master Synchronous 3 D Sv2 Fully Authenticated Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>11964325054750 1c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>5555555555559997</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>02</eci>
     <cavv>kQOImea0C/2ta1NZl4Hhwslmomqj</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0a02e1b0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0a02e250-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
 </payment_transaction>'
```

Visa Synchronous 3 D Sv2 Attempted Authentication Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>11964325054750 1c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4012000000060085</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>06</eci>
     <cavv>MDAwMDAwMDAwMDAxMTA2Njk5NFg=</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0a0b6830-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0a0b6c40-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
   <sca_params>
     <exemption>trusted_merchant</exemption>
     <visa_merchant_id>00000000</visa_merchant_id>
   </sca_params>
 </payment_transaction>'
```

Master Synchronous 3 D Sv2 Attempted Authentication Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>5555555555559997</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>01</eci>
     <cavv>kE0Imea0C/2ta1NZl4Hhwslmomqj</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0a146bc0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0a146c70-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
 </payment_transaction>'
```

Master Synchronous 3 D Sv2 Acquirer Exemption Accepted (Tra Already Performed) Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>5169750000001111</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>06</eci>
     <cavv>kNMAAAAA3S13awBkrWtTWZeBBCmy</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0a1c64d0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0a1c65d0-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
   <sca_params>
     <exemption>low_risk</exemption>
   </sca_params>
 </payment_transaction>'
```

Visa Synchronous 3 D Sv2 Acquirer Exemption Accepted (Tra Already Performed) Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4378510000000004</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
```

```
  <business_attributes>
    <event_start_date>27-03-2025</event_start_date>
    <event_end_date>08-04-2025</event_end_date>
    <event_organizer_id>20192375</event_organizer_id>
    <event_id>1912</event_id>
  </business_attributes>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <mpi_params>
    <eci>07</eci>
    <cavv>ApkCAlgGQCECEBJWNgZAAAAAAAA=</cavv>
    <protocol_version>2</protocol_version>
    <protocol_sub_version>2</protocol_sub_version>
    <directory_server_id>0a23c850-d752-013d-a6e2-0a58a9feac02</directory_server_id>
    <acs_transaction_id>0a23c910-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
    <threeds_challenge_indicator>preference</threeds_challenge_indicator>
  </mpi_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
 </payment_transaction>'
```

Asynchronous 3 D Sv2 Frictionless No 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
  <transaction_type>authorize3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4012000000060085</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
```

```
  </payment_transaction>'
```

Asynchronous 3 D Sv2 Frictionless With 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4066330000000004</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <threeds_v2_params>
     <threeds_method>
       <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
     </threeds_method>
     <control>
       <device_type>browser</device_type>
       <challenge_window_size>full_screen</challenge_window_size>
       <challenge_indicator>preference</challenge_indicator>
     </control>
     <purchase>
       <category>service</category>
     </purchase>
     <merchant_risk>
       <shipping_indicator>verified_address</shipping_indicator>
       <delivery_timeframe>electronic</delivery_timeframe>
       <reorder_items_indicator>reordered</reorder_items_indicator>
       <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
       <pre_order_date>27-03-2025</pre_order_date>
       <gift_card>true</gift_card>
       <gift_card_count>2</gift_card_count>
     </merchant_risk>
     <card_holder_account>
       <creation_date>27-02-2024</creation_date>
       <update_indicator>more_than_60days</update_indicator>
       <last_change_date>27-11-2024</last_change_date>
       <password_change_indicator>no_change</password_change_indicator>
       <password_change_date>12-02-2025</password_change_date>
       <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
       <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
       <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
       <transactions_activity_previous_year>10</transactions_activity_previous_year>
       <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
       <purchases_count_last_6_months>5</purchases_count_last_6_months>
       <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
       <registration_indicator>30_to_60_days</registration_indicator>
       <registration_date>27-02-2023</registration_date>
     </card_holder_account>
     <browser>
       <accept_header>*/*</accept_header>
       <java_enabled>false</java_enabled>
       <language>en-GB</language>
       <color_depth>24</color_depth>
       <screen_height>900</screen_height>
       <screen_width>1440</screen_width>
       <time_zone_offset>-120</time_zone_offset>
       <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
     </browser>
     <sdk>
       <interface>native</interface>
       <ui_types>
         <ui_type>multi_select</ui_type>
       </ui_types>
       <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
       <encrypted_data>encrypted-data-here</encrypted_data>
       <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
       <max_timeout>10</max_timeout>
       <reference_number>sdk-reference-number-here</reference_number>
     </sdk>
   </threeds_v2_params>
   <sca_params>
     <exemption>low_risk</exemption>
   </sca_params>
 </payment_transaction>'
```

Asynchronous 3 D Sv2 Challenge No 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4918190000000002</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
```

```
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```

Asynchronous 3 D Sv2 Challenge With 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>authorize3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4938730000000001</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
```

```
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```


**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **authorize3d** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| preauthorization | optional | "true" | Signifies whether a preauthorization transaction is performed. Check the Preauthorizations section or contact tech support for more details. |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| crypto | optional | "true" | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| notification_url | required[1] | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required[1] | url | URL where customer is sent to after successful payment |
| return_failure_url | required[1] | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount to act as an account verification transaction - Contact tech-support@e-comprocessing.com for more details regarding this scenario. |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(45) | Full name of customer as printed on credit card (first name and last name at least). Note, for async 3DSv2 transactions, the card holder name must NOT contain more than **45** chars. Otherwise, the rest will be truncated in the authentication request. |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| scheme_tokenized | required* | "true" | Required when the `card_number` is DPAN instead of Funding Primary Account Number, see Tokenized e-commerce for details |
| recurring_type | optional | string(255) | Specifies recurring type of the transaction, can be 'initial' or 'managed'. |
| installment_plan_id | optional | string | The ID of the chosen installment plan. Check Installments for more details. |
| installment_plan_reference | optional | string | The installment plan reference identifier returned by the schemes. |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(16) | Must contain a valid international phone number of the customer as per the **ITU-T E.164**. It's recommended to not submit a customer phone number containing more than **15** digits or less than **7** digits. Note, for async 3DS transactions that are using the 3DSv2 authentication protocol, it |

| | | | |
|---|---|---|---|
| | | | will be shortened up to **15** digits and a prefix **+** for international phone number will be added if missing. |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(150) | Primary address. The field length is limited to **150** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string(16) | The field that holds the zip code is limited to **16** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| city | required* | string(50) | The field that holds the city is limited to **50** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(3) | The field that holds the country state is limited to **3** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. Note: The value should be the country subdivision code defined in ISO 3166-2. Invalid values submitted NOT as per the ISO might lead to data integrity issues. |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(150) | Primary address. The field length is limited to **150** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string(16) | The field that holds the zip code is limited to **16** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| city | optional | string(50) | The field that holds the city is limited to **50** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(3) | The field that holds the country state is limited to **3** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. Note: The value should be the country subdivision code defined in ISO 3166-2. Invalid values submitted NOT as per the ISO might lead to data integrity issues. |
| country | optional | string(2) | Country code in ISO 3166 |
| **mpi_params** | required[2] | | |
| cavv | required[3] | string(255) | Verification Id of the authentication. Please note this can be the CAVV for Visa Card or UCAF to identify MasterCard. |
| eci | required[3] | string(255) | See Electronic Commerce Indicator as returned from the MPI for details |
| protocol_version | required[4] | string | The used 3DS protocol version. |
| protocol_sub_version | required[4] | string | The used 3DS protocol sub-version. |
| directory_server_id | required[4] | string | The Directory Server ID used for 3DSecure transactions through the 3DSv2 authentication protocol. |
| acs_transaction_id | optional | string | The ACS Transaction ID and is optional for 3DS transactions, but highly recommended for increasing the approval ratio. |
| threeds_challenge_indicator | optional | string | The 3DS challenge indicator that represents the exact indicator used during the authentication request to the MPI provider for synchronous 3DS transactions. It is optional but highly recommended for increasing the approval ratio. It can only contain one of the following values no_preference, no_challenge_requested, preference and mandate. The default value is no_preference. Check 3DS Challenge Indicators for more details. |
| **sca_params** | optional | | SCA params |
| exemption | optional | string | The exemption that the transaction should take advantage of. Note that the requested exemption may not be accepted due to internal risk validations. Check SCA exemption values. |
| visa_merchant_id | required[5] | string(8) | VMID assigned by Visa if participating in Trusted merchant program. |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |
| **sender** | optional | | Funding Transaction Sender details |

| | | | |
|---|---|---|---|
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |
| **threeds_v2_params** | required* | | 3DSv2 async parameters. They must be submitted in order to use the 3DSv2 authentication protocol in asynchronous workflow |
| **threeds_method** | optional | | 3DS-Method related parameters for any callbacks and notifications. |
| callback_url | optional | url | Specific 3DS-Method callback URL after the 3DS-Method completes. The actual status will be provided via HTTP POST to that URL. For more information, go to 3DSv2 method params |
| **control** | required* | | General params for preferences in authentication flow and providing device interface information. |
| device_type | required* | string | Identifies the device channel of the consumer, **required** in the 3DSv2 authentication protocol. For more information, go to 3DSv2 control params |
| challenge_window_size | required* | string | Identifies the size of the challenge window for the consumer. For more information, go to 3DSv2 control params |
| challenge_indicator | optional | string | The value has weight and might impact the decision whether a challenge will be required for the transaction or not. If not provided, it will be interpreted as **no_preference**. For more information, go to 3DSv2 control params |
| **purchase** | optional | | Purchase related params providing with additional information regarding the order. |
| category | optional | string | Identifies the type of transaction being authenticated. This field is required in some markets.Accepted values are: **goods**, **service**, **check_acceptance**, **account_funding**, **quasi_cash**, **prepaid_activation**, **loan**. |
| **merchant_risk** | recommended | | Merchant risk assessment params. They are all optional, but recommended. |
| shipping_indicator | optional | string(16) | Indicator code that most accurately describes the shipping method for the cardholder specific transaction. If one or more items are included in the sale, use the Shipping Indicator code for the physical goods. If all digital goods, use the code that describes the most expensive item. Accepted values are: **same_as_billing**, **stored_address**, **verified_address**, **pick_up**, **digital_goods**, **travel**, **event_tickets**, **other**. |
| delivery_timeframe | optional | string(11) | Indicates the merchandise delivery timeframe. Accepted values are: **electronic**, **same_day**, **over_night**, **another_day**. |
| reorder_items_indicator | optional | string(10) | Indicates whether the cardholder is reordering previously purchased merchandise. Accepted values are: **first_time**, **reordered**. |
| pre_order_purchase_indicator | optional | string(21) | Indicates whether cardholder is placing an order for merchandise with a future-availability or release date. Accepted values are: **merchandise_available**, **future_availability**. |
| pre_order_date | optional | dd-mm-yyyy | For a pre-ordered purchase, the expected date that the merchandise will be available. |
| gift_card | optional | 'true' | Prepaid or gift card purchase. |
| gift_card_count | optional | integer | For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased. The value is limited to **99**. |
| **card_holder_account** | recommended | | Cardholder account additional information. They are all optional, but recommended, because they have a significant impact on approval rates |
| creation_date | optional | dd-mm-yyyy | Date that the cardholder opened the account with the 3DS Requester. |
| update_indicator | optional | string(19) | Length of time since the cardholder’s account information with the 3DS Requestor was last changed. Includes Billing or Shipping address, new payment account, or new user(s) added. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| last_change_date | optional | dd-mm-yyyy | Date that the cardholder’s account with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment accout, or new user(s) added. |
| password_change_indicator | optional | string(18) | Length of time since the cardholder account with the 3DS Requestor had a password change or account reset. Accepted values are: **no_change**, **during_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| password_change_date | optional | dd-mm-yyyy | Date that cardholder’s account with the 3DS Requestor had a password change or account reset. |
| shipping_address_usage_indicator | optional | string(19) | Indicates when the shipping address used for this transaction was first used with the 3DS Requestor. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| shipping_address_date_first_used | optional | dd-mm-yyyy | Date when the shipping address used for this transaction was first used with the 3DS Requestor. |
| transactions_activity_last_24_hours | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours. |
| transactions_activity_previous_year | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year. |
| provision_attempts_last_24_hours | optional | integer | Number of Add Card attempts in the last 24 hours. |
| purchases_count_last_6_months | optional | integer | Number of purchases with this cardholder account during the previous six months. |
| suspicious_activity_indicator | optional | string(22) | Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account. Accepted values are: **no_suspicious_observed**, **suspicious_observed**. |
| registration_indicator | optional | string(19) | Indicates the length of time that the payment account was enrolled in the cardholder’s account with the 3DS Requester. Accepted values are: **guest_checkout**, **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| registration_date | optional | dd-mm-yyyy | Date that the payment account was enrolled in the cardholder’s account with the 3DS Requestor. |
| **browser** | required* | | For browser-based transactions. They are all *required* in case the **device_type** is set to **browser** |
| accept_header | required* | string(2048) | The exact content of the HTTP **ACCEPT** header as sent to the 3DS Requester from the Cardholder browser. Any other header different than the **ACCEPT** header will be rejected. Example: `application/json,text/plain,text/html,*/*`. |
| java_enabled | required* | boolean | Boolean that represents the ability of the cardholder browser to execute Java. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.javaEnabled`. |
| language | required* | string(8) | Value representing the browser language as defined in IETF BCP47. Note that only one browser language tag is about to be submitted as per the above **IETF BCP47**. Numeric chars are also allowed in the subtag and will represent the region.Example: `en-GB`, `zh-guoyu`, `fil-PH`, `gsw`, `es-419`, `de-1996`, etc. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.language`. |
| color_depth | required* | integer | Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from Cardholder browser using the `screen.colorDepth` property. The value as per EMVCo specs can be one of **1**, **4**, **8**, **15**, **16**, **24**, **32**, **48**. In case, an unsupported `color_depth` is determined, the nearest supported value that is less than the actual one needs to be submitted. For example, if the obtained value is **30**, which is not supported as per EMVCo specs, **24** has to be submitted. |
| screen_height | required* | integer | Total height of the Cardholder's screen in pixels. Value is returned from the screen.height property. |
| screen_width | required* | integer | Total width of the Cardholder's screen in pixels. Value is returned from the screen.width property. |
| time_zone_offset | required* | string(5) | Time difference between UTC time and the Cardholder browser local time, in **minutes**. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. If **UTC -5** hours then submit `300` or `+300`, If **UTC +2** hours then `-120`. The value can be retrieved using Javascript `getTimezoneOffset()` method over **Date** object. |
| user_agent | required* | string(2048) | Exact content of the HTTP user-agent header. |
| **sdk** | required* | | For application-based transactions. They are all *required* in case the **device_type** is set to **application** |

| | | | |
|---|---|---|---|
| interface | required* | string(6) | SDK Interface types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. Accepted values are: **native**, **html**, **both**. |
| **ui_types** | required* | | Lists all UI types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. |
| ui_type | required* | string(13) | UI type that the device of the consumer supports for displaying specific challenge interface. Accepted values are: **text**, **single_select**, **multi_select**, **out_of_bag**, **other_html**. |
| application_id | required* | string(36) | Universally unique ID created upon all installations and updates of the 3DS Requestor APp on a Customer Device. This will be newly generated and stored by the 3DS SDK for each installation or update. The field is limited to 36 characters and it shall have a canonical format as defined in IETF RFC 4122. |
| encrypted_data | required* | string(64000) | JWE Object as defined Section 6.2.2.1 containing data encrypted by the SDK for the DS to decrypt. The data will be present when sending to DS, but not present from DS to ACS. |
| ephemeral_public_key_pair | required* | string(256) | Public key component of the ephemeral key pair generated by the 3DS SDK and used to establish session keys between the 3DS SDK and ACS. In AReq, this data element is contained within the ACS Signed Content JWS Object. The field is limited to maximum 256 characters. |
| max_timeout | required* | integer | Indicates the maximum amount of time (in minutes) for all exchanges. The field shall have value greater or equals than 05. |
| reference_number | required* | string(32) | Identifies the vendor and version of the 3DS SDK that is integrated in a 3DS Requestor App, assigned by EMVCo when the 3DS SDK is approved. The field is limited to 32 characters. |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

**1** - Required if `mpi_params` is not present, the transaction will be handled asynchronously. Not required if configured on Terminal or Merchant level. Contact Tech Support for more details.

**2** - Required if transaction should be handled synchronous.

**3** - `eci` is always required if `mpi_params` is present.

`cavv` is not required for the 3D attempted only workflow, but it is strongly recommended in a combination with the Directory Server ID in order to be in the scope of the 3DSv2 authentication protocol.

**4** - `protocol_version` is required due to the only one 3DSv2 authentication protocol that is currently supported.

`directory_server_id` is mandatory when protocol_version is 2. May be omitted for scheme tokenized transactions.

**5** - `visa_merchant_id` is required when exemption value is `trusted_merchant`.

Frictionless / Challenge Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize3d</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <threeds_method_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method</threeds_method_url>
  <threeds_method_continue_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method/44177a21403427eb96664a6d7e5d5d48</threeds_method_continue_url>
  <timestamp>2025-02-26T16:01:34Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```


Challenge Without 3 Ds Method Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize3d</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <redirect_url>https://staging.gate.e-comprocessing.net/threeds/authentication/44177a21403427eb96664a6d7e5d5d48</redirect_url>
  <redirect_url_type>3ds_v2_challenge</redirect_url_type>
  <timestamp>2025-02-26T16:01:34Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| payment_account_reference | string(255) | Payment Account Reference value returned from the schemes. Links tokenized and PAN-based transactions. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where the consumer has to be redirected to complete the payment process unless a 3DSecure Method is required. This **redirect_url** will not be included in the response if a 3DS-Method submission is required. For more information, to go 3DSv2 authentication flows |
| redirect_url_type | string(64) | The type of the redirect URL in the 3DS scope. It will be present only for asynchronous 3D transactions when an interaction between consumer and issuer is required. This type identifies what kind of redirect url is returned, namely 3DSv2 Challenge. For more information, to go 3DSv2 authentication flows |
| threeds_method_url | url | 3DSecure Method URL. It will be present only then 3DS-Method is required for 3D transaction. A 3DS-Method submission inside an iframe is required to be submitted using HTTP POST. For more information, to go 3DSv2 authentication flows |
| threeds_method_continue_url | url | This is an API endpoint that accepts HTTP PUT & HTTP PATCH requests. It will be present when the **threeds_method_url** is included in the response. An HTTP PUT request must be submitted to that endpoint together with the proper signature to determine what the next step in the authentication is. For more information, to go 3DSv2 authentication flows |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to 'true' if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_transaction_link_id | string(22) | The transaction unique identifier returned from the schemes. Corresponds to ADDITIONAL SERVICE DATA (field 65, MasterCard Transaction Link Id [TLID]). |
| scheme_pan_indicator | string(1) | The account number indicator in scheme-tokenized transactions. See Account number indicator codes for details. |
| scheme_pan_tail | string(4) | The tail of the PAN number tail associated with the token in scheme-tokenized transactions. Returned in conjunction with `scheme_pan_indicator`. |
| scheme_max_settlement_date | string(10) | The final date an authorization could be captured which is returned from the card scheme. See Scheme Maximum Settlement Date for more details. |
| scheme_authentication_data_quality | boolean | The Authentication Data Quality Indicator returned by the card scheme is used to indicate whether a transaction meets the authentication data quality requirements. The value can be either `true` or `false`. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |
| **threeds** | | |
| eci | string(2) | See Electronic Commerce Indicator as returned from the MPI for details |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize3d</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <technical_message>expiration_year is invalid</technical_message>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:34Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| **threeds** | | |
| **authentication** | | |
| status_reason_code | string(2) | See 3DS Authentication Status Reason Codes for details. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize3d</transaction_type>
  <status>declined</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>600</code>
  <technical_message>Cardholder not participating 3DS.</technical_message>
  <message>Transaction failed, please contact support!</message>
  <timestamp>2025-02-26T16:01:34Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <threeds>
    <eci>06</eci>
    <authentication>
      <status_reason_code>08</status_reason_code>
    </authentication>
  </threeds>
</payment_response>
```

**SALE 3D**

Sale3D transactions basically have the same request as standard sale transactions.

Sale3D transactions can be handled **synchronous** or **asynchronous** depending on the parameers passed. If mpi_params is passed the workflow will be **synchronous**. If **notification_url**, **return_success_url** and **return_failure_url** are passed the workflow will be **asynchronous**.

This transaction type supports Tokenization.

This transaction type supports Level 3 travel data.

This transaction type could require business attributes.

An exemption from Strong Customer Authentication (SCA) can be requested by submitting an **exemption** with `low_risk` under SCA params.

In case the issuer accepts the exemption, a step up in the authentication flow might not be required because the transaction's risk analysis has already been performed by acquirer.

Note, the requested exemption might not be accepted due to internal risk validations.

For example, to be able to utilize the low risk exemption, the BIN country of the card must be part of the European Economic Area (EEA).

Furthermore, the acquirer could accept the merchant low-risk exemption request only if the transaction amount does not exceed the acquirer low-risk exemption threshold.

Finally, the ACS might not acknowledge the merchant/acquirer's exemption request and may still require a step up in the cardholder authentication.

This transaction type supports Installments.

Visa Synchronous 3 D Sv2 Fully Authenticated Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4012000000060085</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>05</eci>
     <cavv>MDAwMDAwMDAwMDAxMTA2Njk5NFg=</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0a813b00-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0a813d10-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
   <sca_params>
     <exemption>trusted_merchant</exemption>
     <visa_merchant_id>00000000</visa_merchant_id>
   </sca_params>
 </payment_transaction>'
```

Master Synchronous 3 D Sv2 Fully Authenticated Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>5555555555559997</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>02</eci>
     <cavv>kBOImea0C/2ta1NZl4Hhwslmomqj</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0a878a90-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0a878b40-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
 </payment_transaction>'
```

Visa Synchronous 3 D Sv2 Attempted Authentication Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
```

```
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4012000000060085</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>06</eci>
     <cavv>MDAwMDAwMDAwMDAxMTA2Njk5NFg=</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0a8f92c0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0a8f93d0-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
   <sca_params>
     <exemption>trusted_merchant</exemption>
     <visa_merchant_id>00000000</visa_merchant_id>
   </sca_params>
 </payment_transaction>'
```

Master Synchronous 3 D Sv2 Attempted Authentication Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>5555555555559997</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>01</eci>
     <cavv>kEOImea0C/2ta1NZl4Hhwslmomqj</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0a9a4aa0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0a9a4b50-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
 </payment_transaction>'
```

Master Synchronous 3 D Sv2 Acquirer Exemption Accepted (Tra Already Performed) Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>5169750000001111</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
```

```
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <mpi_params>
    <eci>06</eci>
    <cavv>kNMAAAAA3S13awBkrWtTWZeBBCmy</cavv>
    <protocol_version>2</protocol_version>
    <protocol_sub_version>2</protocol_sub_version>
    <directory_server_id>0aa18d20-d752-013d-a6e2-0a58a9feac02</directory_server_id>
    <acs_transaction_id>0aa18dd0-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
    <threeds_challenge_indicator>preference</threeds_challenge_indicator>
  </mpi_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```

Visa Synchronous 3 D Sv2 Acquirer Exemption Accepted (Tra Already Performed) Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>sale3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4378510000000004</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <business_attributes>
    <event_start_date>27-03-2025</event_start_date>
    <event_end_date>08-04-2025</event_end_date>
    <event_organizer_id>20192375</event_organizer_id>
    <event_id>1912</event_id>
  </business_attributes>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <mpi_params>
    <eci>07</eci>
    <cavv>ApkCAlgGQCECEBJWNgZAAAAAAAA=</cavv>
    <protocol_version>2</protocol_version>
    <protocol_sub_version>2</protocol_sub_version>
    <directory_server_id>0aa8bf30-d752-013d-a6e2-0a58a9feac02</directory_server_id>
    <acs_transaction_id>0aa8bfe0-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
    <threeds_challenge_indicator>preference</threeds_challenge_indicator>
  </mpi_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```

Asynchronous 3 D Sv2 Frictionless No 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>sale3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4012000000060085</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
```

```
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```

Asynchronous 3 D Sv2 Frictionless With 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>sale3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4066330000000004</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
```

```
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```

Asynchronous 3 D Sv2 Challenge No 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>sale3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4918190000000002</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```

Asynchronous 3 D Sv2 Challenge With 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
```

```
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>sale3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4938730000000001</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sale3d** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| crypto | optional | "true" | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| notification_url | required[1] | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required[1] | url | URL where customer is sent to after successful payment |
| return_failure_url | required[1] | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount to act as an account verification transaction - Contact tech-support@e-comprocessing.com for more details regarding this scenario. |

| | | | |
|---|---|---|---|
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(45) | Full name of customer as printed on credit card (first name and last name at least). Note, for async 3DSv2 transactions, the card holder name must NOT contain more than **45** chars. Otherwise, the rest will be truncated in the authentication request. |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| scheme_tokenized | required* | "true" | Required when the `card_number` is DPAN instead of Funding Primary Account Number, see Tokenized e-commerce for details |
| recurring_type | optional | string(255) | Specifies recurring type of the transaction, can be 'initial' or 'managed'. |
| installment_plan_id | optional | string | The ID of the chosen installment plan. Check Installments for more details. |
| installment_plan_reference | optional | string | The installment plan reference identifier returned by the schemes. |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(16) | Must contain a valid international phone number of the customer as per the **ITU-T E.164**. It's recommended to not submit a customer phone number containing more than **15** digits or less than **7** digits. Note, for async 3DS transactions that are using the 3DSv2 authentication protocol, it will be shortened up to **15** digits and a prefix **+** for international phone number will be added if missing. |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(150) | Primary address. The field length is limited to **150** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string(16) | The field that holds the zip code is limited to **16** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| city | required* | string(50) | The field that holds the city is limited to **50** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(3) | The field that holds the country state is limited to **3** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. Note: The value should be the country subdivision code defined in ISO 3166-2. Invalid values submitted NOT as per the ISO might lead to data integrity issues. |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(150) | Primary address. The field length is limited to **150** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string(16) | The field that holds the zip code is limited to **16** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| city | optional | string(50) | The field that holds the city is limited to **50** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(3) | The field that holds the country state is limited to **3** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. Note: The value should be the country subdivision code defined in ISO 3166-2. Invalid values submitted NOT as per the ISO might lead to data integrity issues. |
| country | optional | string(2) | Country code in ISO 3166 |
| **mpi_params** | required[2] | | |
| cavv | required[3] | string(255) | Verification Id of the authentication. Please note this can be the CAVV for Visa Card or UCAF to identify MasterCard. |
| eci | required[3] | string(255) | See Electronic Commerce Indicator as returned from the MPI for details |
| protocol_version | required[4] | string | The used 3DS protocol version. |

| | | | |
|---|---|---|---|
| protocol_sub_version | required[4] | string | The used 3DS protocol sub-version. |
| directory_server_id | required[4] | string | The Directory Server ID used for 3DSecure transactions through the 3DSv2 authentication protocol. |
| acs_transaction_id | optional | string | The ACS Transaction ID and is optional for 3DS transactions, but highly recommended for increasing the approval ratio. |
| threeds_challenge_indicator | optional | string | The 3DS challenge indicator that represents the exact indicator used during the authentication request to the MPI provider for synchronous 3DS transactions. It is optional but highly recommended for increasing the approval ratio. It can only contain one of the following values no_preference, no_challenge_requested, preference and mandate. The default value is no_preference. Check 3DS Challenge Indicators for more details. |
| **sca_params** | optional | | SCA params |
| exemption | optional | string | The exemption that the transaction should take advantage of. Note that the requested exemption may not be accepted due to internal risk validations. Check SCA exemption values. |
| visa_merchant_id | required[5] | string(8) | VMID assigned by Visa if participating in Trusted merchant program. |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |
| **sender** | optional | | Funding Transaction Sender details |
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |
| **threeds_v2_params** | required* | | 3DSv2 async parameters. They must be submitted in order to use the 3DSv2 authentication protocol in asynchronous workflow |
| **threeds_method** | optional | | 3DS-Method related parameters for any callbacks and notifications. |
| callback_url | optional | url | Specific 3DS-Method callback URL after the 3DS-Method completes. The actual status will be provided via HTTP POST to that URL. For more information, go to 3DSv2 method params |
| **control** | required* | | General params for preferences in authentication flow and providing device interface information. |
| device_type | required* | string | Identifies the device channel of the consumer, **required** in the 3DSv2 authentication protocol. For more information, go to 3DSv2 control params |
| challenge_window_size | required* | string | Identifies the size of the challenge window for the consumer. For more information, go to 3DSv2 control params |
| challenge_indicator | optional | string | The value has weight and might impact the decision whether a challenge will be required for the transaction or not. If not provided, it will be interpreted as **no_preference**. For more information, go to 3DSv2 control params |
| **purchase** | optional | | Purchase related params providing with additional information regarding the order. |
| category | optional | string | Identifies the type of transaction being authenticated. This field is required in some markets.Accepted values are: **goods**, **service**, **check_acceptance**, **account_funding**, **quasi_cash**, **prepaid_activation**, **loan**. |
| **merchant_risk** | recommended | | Merchant risk assessment params. They are all optional, but recommended. |
| shipping_indicator | optional | string(16) | Indicator code that most accurately describes the shipping method for the cardholder specific transaction. If one or more items are included in the sale, use the Shipping Indicator code for the physical goods. If all digital goods, use the code that describes the most expensive item. Accepted values are: **same_as_billing**, **stored_address**, **verified_address**, **pick_up**, **digital_goods**, **travel**, **event_tickets**, **other**. |
| delivery_timeframe | optional | string(11) | Indicates the merchandise delivery timeframe. Accepted values are: **electronic**, **same_day**, **over_night**, **another_day**. |
| reorder_items_indicator | optional | string(10) | Indicates whether the cardholder is reordering previously purchased merchandise. Accepted values are: **first_time**, **reordered**. |
| pre_order_purchase_indicator | optional | string(21) | Indicates whether cardholder is placing an order for merchandise with a future-availability or release date. Accepted values are: **merchandise_available**, **future_availability**. |
| pre_order_date | optional | dd-mm-yyyy | For a pre-ordered purchase, the expected date that the merchandise will be available. |
| gift_card | optional | 'true' | Prepaid or gift card purchase. |
| gift_card_count | optional | integer | For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased. The value is limited to **99**. |
| **card_holder_account** | recommended | | Cardholder account additional information. They are all optional, but recommended, because they have a significant impact on approval rates |
| creation_date | optional | dd-mm-yyyy | Date that the cardholder opened the account with the 3DS Requester. |
| update_indicator | optional | string(19) | Length of time since the cardholder’s account information with the 3DS Requestor was last changed. Includes Billing or Shipping address, new payment account, or new user(s) added. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| last_change_date | optional | dd-mm-yyyy | Date that the cardholder’s account with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment accout, or new user(s) added. |
| password_change_indicator | optional | string(18) | Length of time since the cardholder account with the 3DS Requestor had a password change or account reset. Accepted values are: **no_change**, **during_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| password_change_date | optional | dd-mm-yyyy | Date that cardholder’s account with the 3DS Requestor had a password change or account reset. |
| shipping_address_usage_indicator | optional | string(19) | Indicates when the shipping address used for this transaction was first used with the 3DS Requestor. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| shipping_address_date_first_used | optional | dd-mm-yyyy | Date when the shipping address used for this transaction was first used with the 3DS Requestor. |
| transactions_activity_last_24_hours | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours. |
| transactions_activity_previous_year | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year. |

| | | | |
|---|---|---|---|
| provision_attempts_last_24_hours | optional | integer | Number of Add Card attempts in the last 24 hours. |
| purchases_count_last_6_months | optional | integer | Number of purchases with this cardholder account during the previous six months. |
| suspicious_activity_indicator | optional | string(22) | Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account. Accepted values are: **no_suspicious_observed**, **suspicious_observed**. |
| registration_indicator | optional | string(19) | Indicates the length of time that the payment account was enrolled in the cardholder's account with the 3DS Requester. Accepted values are: **guest_checkout**, **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| registration_date | optional | dd-mm-yyyy | Date that the payment account was enrolled in the cardholder's account with the 3DS Requestor. |
| **browser** | required* | | For browser-based transactions. They are all *required* in case the **device_type** is set to **browser** |
| accept_header | required* | string(2048) | The exact content of the HTTP **ACCEPT** header as sent to the 3DS Requester from the Cardholder browser. Any other header different than the **ACCEPT** header will be rejected. Example: `application/json,text/plain,text/html,*/*`. |
| java_enabled | required* | boolean | Boolean that represents the ability of the cardholder browser to execute Java. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.javaEnabled`. |
| language | required* | string(8) | Value representing the browser language as defined in IETF BCP47. Note that only one browser language tag is about to be submitted as per the above **IETF BCP47**. Numeric chars are also allowed in the subtag and will represent the region.Example: `en-GB`, `zh-guoyu`, `fil-PH`, `gsw`, `es-419`, `de-1996`, etc. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.language`. |
| color_depth | required* | integer | Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from Cardholder browser using the `screen.colorDepth` property. The value as per EMVCo specs can be one of **1**, **4**, **8**, **15**, **16**, **24**, **32**, **48**. In case, an unsupported `color_depth` is determined, the nearest supported value that is less than the actual one needs to be submitted. For example, if the obtained value is **30**, which is not supported as per EMVCo specs, **24** has to be submitted. |
| screen_height | required* | integer | Total height of the Cardholder's screen in pixels. Value is returned from the screen.height property. |
| screen_width | required* | integer | Total width of the Cardholder's screen in pixels. Value is returned from the screen.width property. |
| time_zone_offset | required* | string(5) | Time difference between UTC time and the Cardholder browser local time, in **minutes**. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. If **UTC -5** hours then submit `300` or `+300`, If **UTC +2** hours then `-120`. The value can be retrieved using Javascript `getTimezoneOffset()` method over **Date** object. |
| user_agent | required* | string(2048) | Exact content of the HTTP user-agent header. |
| **sdk** | required* | | For application-based transactions. They are all *required* in case the **device_type** is set to **application** |
| interface | required* | string(6) | SDK Interface types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. Accepted values are: **native**, **html**, **both**. |
| **ui_types** | required* | | Lists all UI types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. |
| ui_type | required* | string(13) | UI type that the device of the consumer supports for displaying specific challenge interface. Accepted values are: **text**, **single_select**, **multi_select**, **out_of_bag**, **other_html**. |
| application_id | required* | string(36) | Universally unique ID created upon all installations and updates of the 3DS Requestor APp on a Customer Device. This will be newly generated and stored by the 3DS SDK for each installation or update. The field is limited to 36 characters and it shall have a canonical format as defined in IETF RFC 4122. |
| encrypted_data | required* | string(64000) | JWE Object as defined Section 6.2.2.1 containing data encrypted by the SDK for the DS to decrypt. The data will be present when sending to DS, but not present from DS to ACS. |
| ephemeral_public_key_pair | required* | string(256) | Public key component of the ephemeral key pair generated by the 3DS SDK and used to establish session keys between the 3DS SDK and ACS. In AReq, this data element is contained within the ACS Signed Content JWS Object. The field is limited to maximum 256 characters. |
| max_timeout | required* | integer | Indicates the maximum amount of time (in minutes) for all exchanges. The field shall have value greater or equals than 05. |
| reference_number | required* | string(32) | Identifies the vendor and version of the 3DS SDK that is integrated in a 3DS Requestor App, assigned by EMVCo when the 3DS SDK is approved. The field is limited to 32 characters. |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

**1** - Required if `mpi_params` is not present, the transaction will be handled asynchronously. Not required if configured on Terminal or Merchant level. Contact Tech Support for more details.

**2** - Required if transaction should be handled synchronous.

**3** - `eci` is always required if `mpi_params` is present.

`cavv` is not required for the 3D attempted only workflow, but it is strongly recommended in a combination with the Directory Server ID in order to be in the scope of the 3DSv2 authentication protocol.

**4** - `protocol_version` is required due to the only one 3DSv2 authentication protocol that is currently supported.

`directory_server_id` is mandatory when protocol_version is 2. May be omitted for scheme tokenized transactions.

**5** - `visa_merchant_id` is required when exemption value is `trusted_merchant`.

Frictionless / Challenge Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
```

```
  <transaction_type>sale3d</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <threeds_method_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method</threeds_method_url>
  <threeds_method_continue_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method/44177a21403427eb96664a6d7e5d5d48</threeds_method_continue_url>
  <timestamp>2025-02-26T16:01:34Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```


Challenge Without 3 Ds Method Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <redirect_url>https://staging.gate.e-comprocessing.net/threeds/authentication/44177a21403427eb96664a6d7e5d5d48</redirect_url>
  <redirect_url_type>3ds_v2_challenge</redirect_url_type>
  <timestamp>2025-02-26T16:01:34Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| payment_account_reference | string(255) | Payment Account Reference value returned from the schemes. Links tokenized and PAN-based transactions. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where the consumer has to be redirected to complete the payment process unless a 3DSecure Method is required. This **redirect_url** will not be included in the response if a 3DS-Method submission is required. For more information, to go 3DSv2 authentication flows |
| redirect_url_type | string(64) | The type of the redirect URL in the 3DS scope. It will be present only for asynchronous 3D transactions when an interaction between consumer and issuer is required. This type identifies what kind of redirect url is returned, namely 3DSv2 Challenge. For more information, to go 3DSv2 authentication flows |
| threeds_method_url | url | 3DSecure Method URL. It will be present only then 3DS-Method is required for 3D transaction. A 3DS-Method submission inside an iframe is required to be submitted using HTTP POST. For more information, to go 3DSv2 authentication flows |
| threeds_method_continue_url | url | This is an API endpoint that accepts HTTP PUT & HTTP PATCH requests. It will be present when the **threeds_method_url** is included in the response. An HTTP PUT request must be submitted to that endpoint together with the proper signature to determine what the next step in the authentication is. For more information, to go 3DSv2 authentication flows |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ‘true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_transaction_link_id | string(22) | The transaction unique identifier returned from the schemes. Corresponds to ADDITIONAL SERVICE DATA (field 65, MasterCard Transaction Link Id [TLID]). |
| scheme_pan_indicator | string(1) | The account number indicator in scheme-tokenized transactions. See Account number indicator codes for details. |
| scheme_pan_tail | string(4) | The tail of the PAN number tail associated with the token in scheme-tokenized transactions. Returned in conjunction with `scheme_pan_indicator`. |

| | | |
|---|---|---|
| scheme_max_settlement_date | string(10) | The final date an authorization could be captured which is returned from the card scheme. See Scheme Maximum Settlement Date for more details. |
| scheme_authentication_data_quality | boolean | The Authentication Data Quality Indicator returned by the card scheme is used to indicate whether a transaction meets the authentication data quality requirements. The value can be either `true` or `false`. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |
| **threeds** | | |
| eci | string(2) | See Electronic Commerce Indicator as returned from the MPI for details |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <technical_message>expiration_year is invalid</technical_message>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:35Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| **threeds** | | |
| **authentication** | | |
| status_reason_code | string(2) | See 3DS Authentication Status Reason Codes for details. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>declined</status>
  <mode>live</mode>
```

```
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>600</code>
  <technical_message>Cardholder not participating 3DS.</technical_message>
  <message>Transaction failed, please contact support!</message>
  <timestamp>2025-02-26T16:01:35Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <threeds>
    <eci>06</eci>
    <authentication>
      <status_reason_code>08</status_reason_code>
    </authentication>
  </threeds>
</payment_response>
```

**INIT RECURRING SALE 3D**

Init Recurring Sale 3D transanction will be soon deprecated. Please start using Sale 3D or Authorize 3D transaction with initial recurring type instead.

InitRecurringSale3D transactions basically have the same request as standard InitRecurringSale transactions.

InitRecurringSale3D transactions can be handled synchronous or asynchronous depending on the parameters passed. If mpi params is passed the workflow will be synchronous. If notification url, return success url and return failure url are passed the workflow will be asynchronous.

This transaction type supports Tokenization.

This transaction type supports Level 3 travel data.

This transaction type could require business attributes.

An exemption from Strong Customer Authentication (SCA) can be requested by submitting an **exemption** with `low_risk` under SCA params.

In case the issuer accepts the exemption, a step up in the authentication flow might not be required because the transaction's risk analysis has already been performed by acquirer.

Note, the requested exemption might not be accepted due to internal risk validations.

For example, to be able to utilize the low risk exemption, the BIN country of the card must be part of the European Economic Area (EEA).

Furthermore, the acquirer could accept the merchant low-risk exemption request only if the transaction amount does not exceed the acquirer low-risk exemption threshold.

Finally, the ACS might not acknowledge the merchant/acquirer's exemption request and may still require a step up in the cardholder authentication.

This transaction type supports Managed Recurring.

Visa Synchronous 3 D Sv2 Fully Authenticated Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>init_recurring_sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4012000000060085</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>05</eci>
     <cavv>MDAwMDAwMDAwMDAxMTA2Njk5NFg=</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0b175dd0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0b176070-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
   <sca_params>
     <exemption>trusted_merchant</exemption>
     <visa_merchant_id>00000000</visa_merchant_id>
   </sca_params>
   <recurring_category>subscription</recurring_category>
 </payment_transaction>'
```

Master Synchronous 3 D Sv2 Fully Authenticated Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
```

```
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>init_recurring_sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>5555555555559997</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>02</eci>
     <cavv>kQOImea0C/2ta1NZl4Hhwslmomqj</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0b200db0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0b200e60-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
   <recurring_category>subscription</recurring_category>
 </payment_transaction>'
```

Visa Synchronous 3 D Sv2 Attempted Authentication Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>init_recurring_sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4012000000060085</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>06</eci>
     <cavv>MDAwMDAwMDAwMDAxMTA2Njk5NFg=</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
     <directory_server_id>0b2be530-d752-013d-a6e2-0a58a9feac02</directory_server_id>
     <acs_transaction_id>0b2be5d0-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
     <threeds_challenge_indicator>preference</threeds_challenge_indicator>
   </mpi_params>
   <sca_params>
     <exemption>trusted_merchant</exemption>
     <visa_merchant_id>00000000</visa_merchant_id>
   </sca_params>
   <recurring_category>subscription</recurring_category>
 </payment_transaction>'
```

Master Synchronous 3 D Sv2 Attempted Authentication Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>init_recurring_sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>5555555555559997</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
```

```
    </business_attributes>
    <billing_address>
      <first_name>Travis</first_name>
      <last_name>Pastrana</last_name>
      <address1>Muster Str. 12</address1>
      <zip_code>10178</zip_code>
      <city>Los Angeles</city>
      <neighborhood>Hollywood</neighborhood>
      <state>CA</state>
      <country>US</country>
    </billing_address>
    <mpi_params>
      <eci>01</eci>
      <cavv>kFOImea0C/2ta1NZl4Hhwslmomqj</cavv>
      <protocol_version>2</protocol_version>
      <protocol_sub_version>2</protocol_sub_version>
      <directory_server_id>0b3592f0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
      <acs_transaction_id>0b359380-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
      <threeds_challenge_indicator>preference</threeds_challenge_indicator>
    </mpi_params>
    <recurring_category>subscription</recurring_category>
  </payment_transaction>'
```

Master Synchronous 3 D Sv2 Acquirer Exemption Accepted (Tra Already Performed) Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
  <?xml version="1.0" encoding="UTF-8"?>
  <payment_transaction>
    <transaction_type>init_recurring_sale3d</transaction_type>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <usage>40208 concert tickets</usage>
    <remote_ip>245.253.2.12</remote_ip>
    <amount>100</amount>
    <currency>USD</currency>
    <card_holder>Travis Pastrana</card_holder>
    <card_number>5169750000001111</card_number>
    <expiration_month>12</expiration_month>
    <expiration_year>2026</expiration_year>
    <cvv>834</cvv>
    <customer_email>travis@example.com</customer_email>
    <customer_phone>+1987987987987</customer_phone>
    <business_attributes>
      <event_start_date>27-03-2025</event_start_date>
      <event_end_date>08-04-2025</event_end_date>
      <event_organizer_id>20192375</event_organizer_id>
      <event_id>1912</event_id>
    </business_attributes>
    <billing_address>
      <first_name>Travis</first_name>
      <last_name>Pastrana</last_name>
      <address1>Muster Str. 12</address1>
      <zip_code>10178</zip_code>
      <city>Los Angeles</city>
      <neighborhood>Hollywood</neighborhood>
      <state>CA</state>
      <country>US</country>
    </billing_address>
    <mpi_params>
      <eci>06</eci>
      <cavv>kNMAAAAA3S13awBkrWtTWZeBBCmy</cavv>
      <protocol_version>2</protocol_version>
      <protocol_sub_version>2</protocol_sub_version>
      <directory_server_id>0b3aead0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
      <acs_transaction_id>0b3aeb80-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
      <threeds_challenge_indicator>preference</threeds_challenge_indicator>
    </mpi_params>
    <sca_params>
      <exemption>low_risk</exemption>
    </sca_params>
    <recurring_category>subscription</recurring_category>
  </payment_transaction>'
```

Visa Synchronous 3 D Sv2 Acquirer Exemption Accepted (Tra Already Performed) Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
  <?xml version="1.0" encoding="UTF-8"?>
  <payment_transaction>
    <transaction_type>init_recurring_sale3d</transaction_type>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <usage>40208 concert tickets</usage>
    <remote_ip>245.253.2.12</remote_ip>
    <amount>100</amount>
    <currency>USD</currency>
    <card_holder>Travis Pastrana</card_holder>
    <card_number>4378510000000004</card_number>
    <expiration_month>12</expiration_month>
    <expiration_year>2026</expiration_year>
    <cvv>834</cvv>
    <customer_email>travis@example.com</customer_email>
    <customer_phone>+1987987987987</customer_phone>
    <business_attributes>
      <event_start_date>27-03-2025</event_start_date>
      <event_end_date>08-04-2025</event_end_date>
      <event_organizer_id>20192375</event_organizer_id>
      <event_id>1912</event_id>
    </business_attributes>
    <billing_address>
      <first_name>Travis</first_name>
      <last_name>Pastrana</last_name>
      <address1>Muster Str. 12</address1>
      <zip_code>10178</zip_code>
      <city>Los Angeles</city>
      <neighborhood>Hollywood</neighborhood>
      <state>CA</state>
      <country>US</country>
    </billing_address>
    <mpi_params>
      <eci>07</eci>
      <cavv>ApkCAlgGQCECEBJWNgZAAAAAAAA=</cavv>
      <protocol_version>2</protocol_version>
      <protocol_sub_version>2</protocol_sub_version>
      <directory_server_id>0b4149b0-d752-013d-a6e2-0a58a9feac02</directory_server_id>
      <acs_transaction_id>0b414a60-d752-013d-a6e2-0a58a9feac02</acs_transaction_id>
      <threeds_challenge_indicator>preference</threeds_challenge_indicator>
    </mpi_params>
    <sca_params>
      <exemption>low_risk</exemption>
```

```
    </sca_params>
    <recurring_category>subscription</recurring_category>
  </payment_transaction>'
```

Asynchronous 3 D Sv2 Frictionless No 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
    -X POST \
    -H "Content-Type: text/xml" \
    -d '
  <?xml version="1.0" encoding="UTF-8"?>
  <payment_transaction>
    <transaction_type>init_recurring_sale3d</transaction_type>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <usage>40208 concert tickets</usage>
    <remote_ip>245.253.2.12</remote_ip>
    <amount>100</amount>
    <currency>USD</currency>
    <card_holder>Travis Pastrana</card_holder>
    <card_number>4012000000060085</card_number>
    <expiration_month>12</expiration_month>
    <expiration_year>2026</expiration_year>
    <cvv>834</cvv>
    <customer_email>travis@example.com</customer_email>
    <customer_phone>+1987987987987</customer_phone>
    <billing_address>
      <first_name>Travis</first_name>
      <last_name>Pastrana</last_name>
      <address1>Muster Str. 12</address1>
      <zip_code>10178</zip_code>
      <city>Los Angeles</city>
      <neighborhood>Hollywood</neighborhood>
      <state>CA</state>
      <country>US</country>
    </billing_address>
    <notification_url>https://www.example.com/notification</notification_url>
    <return_success_url>http://www.example.com/success</return_success_url>
    <return_failure_url>http://www.example.com/failure</return_failure_url>
    <threeds_v2_params>
      <threeds_method>
        <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
      </threeds_method>
      <control>
        <device_type>browser</device_type>
        <challenge_window_size>full_screen</challenge_window_size>
        <challenge_indicator>preference</challenge_indicator>
      </control>
      <purchase>
        <category>service</category>
      </purchase>
      <recurring>
        <expiration_date>27-08-2025</expiration_date>
        <frequency>30</frequency>
      </recurring>
      <merchant_risk>
        <shipping_indicator>verified_address</shipping_indicator>
        <delivery_timeframe>electronic</delivery_timeframe>
        <reorder_items_indicator>reordered</reorder_items_indicator>
        <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
        <pre_order_date>27-03-2025</pre_order_date>
        <gift_card>true</gift_card>
        <gift_card_count>2</gift_card_count>
      </merchant_risk>
      <card_holder_account>
        <creation_date>27-02-2024</creation_date>
        <update_indicator>more_than_60days</update_indicator>
        <last_change_date>27-11-2024</last_change_date>
        <password_change_indicator>no_change</password_change_indicator>
        <password_change_date>12-02-2025</password_change_date>
        <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
        <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
        <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
        <transactions_activity_previous_year>10</transactions_activity_previous_year>
        <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
        <purchases_count_last_6_months>5</purchases_count_last_6_months>
        <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
        <registration_indicator>30_to_60_days</registration_indicator>
        <registration_date>27-02-2023</registration_date>
      </card_holder_account>
      <browser>
        <accept_header>*/*</accept_header>
        <java_enabled>false</java_enabled>
        <language>en-GB</language>
        <color_depth>24</color_depth>
        <screen_height>900</screen_height>
        <screen_width>1440</screen_width>
        <time_zone_offset>-120</time_zone_offset>
        <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
      </browser>
      <sdk>
        <interface>native</interface>
        <ui_types>
          <ui_type>multi_select</ui_type>
        </ui_types>
        <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
        <encrypted_data>encrypted-data-here</encrypted_data>
        <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
        <max_timeout>10</max_timeout>
        <reference_number>sdk-reference-number-here</reference_number>
      </sdk>
    </threeds_v2_params>
    <sca_params>
      <exemption>low_risk</exemption>
    </sca_params>
    <recurring_category>subscription</recurring_category>
  </payment_transaction>'
```

Asynchronous 3 D Sv2 Frictionless With 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
    -X POST \
    -H "Content-Type: text/xml" \
    -d '
  <?xml version="1.0" encoding="UTF-8"?>
  <payment_transaction>
    <transaction_type>init_recurring_sale3d</transaction_type>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <usage>40208 concert tickets</usage>
    <remote_ip>245.253.2.12</remote_ip>
    <amount>100</amount>
    <currency>USD</currency>
    <card_holder>Travis Pastrana</card_holder>
    <card_number>4066330000000004</card_number>
```

```
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <recurring>
      <expiration_date>27-08-2025</expiration_date>
      <frequency>30</frequency>
    </recurring>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
  <recurring_category>subscription</recurring_category>
</payment_transaction>'
```

Asynchronous 3 D Sv2 Challenge No 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>init_recurring_sale3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4918190000000002</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
```

```
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <recurring>
      <expiration_date>27-08-2025</expiration_date>
      <frequency>30</frequency>
    </recurring>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
  <recurring_category>subscription</recurring_category>
</payment_transaction>'
```

Asynchronous 3 D Sv2 Challenge With 3ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>init_recurring_sale3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4938730000000001</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <recurring>
      <expiration_date>27-08-2025</expiration_date>
      <frequency>30</frequency>
    </recurring>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
```

```
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
  <recurring_category>subscription</recurring_category>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **init_recurring_sale3d** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| notification_url | required[1] | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required[1] | url | URL where customer is sent to after successful payment |
| return_failure_url | required[1] | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount in order not to charge the consumer with the initial recurring, but with the followed RecurringSale transactions only. For more information regarding the use cases and scenario, Contact tech-support@e-comprocessing.com for more details. |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(45) | Full name of customer as printed on credit card (first name and last name at least). Note, for async 3DSv2 transactions, the card holder name must NOT contain more than **45** chars. Otherwise, the rest will be truncated in the authentication request. |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| recurring_category | optional | string | Specifies whether the recurring transaction is a subscription(fixed amount, fixed intervals)or if it is a standing order(varying amount, fixed intervals). The allowed values are `subscription` and `standing_order`. The default value is `subscription` |
| scheme_tokenized | required* | "true" | Required when the `card_number` is DPAN instead of Funding Primary Account Number, see Tokenized e-commerce for details |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(16) | Must contain a valid international phone number of the customer as per the **ITU-T E.164**. It's recommended to not submit a customer phone number containing more than **15** digits or less than **7** digits. Note, for async 3DS transactions that are using the 3DSv2 authentication protocol, it will be shortened up to **15** digits and a prefix **+** for international phone number will be added if missing. |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |

| | | | |
|---|---|---|---|
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(150) | Primary address. The field length is limited to **150** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string(16) | The field that holds the zip code is limited to **16** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| city | required* | string(50) | The field that holds the city is limited to **50** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(3) | The field that holds the country state is limited to **3** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. Note: The value should be the country subdivision code defined in ISO 3166-2. Invalid values submitted NOT as per the ISO might lead to data integrity issues. |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(150) | Primary address. The field length is limited to **150** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string(16) | The field that holds the zip code is limited to **16** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| city | optional | string(50) | The field that holds the city is limited to **50** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(3) | The field that holds the country state is limited to **3** chars only for async 3DS transactions that are using the 3DSv2 authentication protocol. Note: The value should be the country subdivision code defined in ISO 3166-2. Invalid values submitted NOT as per the ISO might lead to data integrity issues. |
| country | optional | string(2) | Country code in ISO 3166 |
| **mpi_params** | required[2] | | |
| cavv | required[3] | string(255) | Verification Id of the authentication. Please note this can be the CAVV for Visa Card or UCAF to identify MasterCard. |
| eci | required[3] | string(255) | See Electronic Commerce Indicator as returned from the MPI for details |
| protocol_version | required[4] | string | The used 3DS protocol version. |
| protocol_sub_version | required[4] | string | The used 3DS protocol sub-version. |
| directory_server_id | required[4] | string | The Directory Server ID used for 3DSecure transactions through the 3DSv2 authentication protocol. |
| acs_transaction_id | optional | string | The ACS Transaction ID and is optional for 3DS transactions, but highly recommended for increasing the approval ratio. |
| threeds_challenge_indicator | optional | string | The 3DS challenge indicator that represents the exact indicator used during the authentication request to the MPI provider for synchronous 3DS transactions. It is optional but highly recommended for increasing the approval ratio. It can only contain one of the following values no_preference, no_challenge_requested, preference and mandate. The default value is no_preference. Check 3DS Challenge Indicators for more details. |
| **sca_params** | optional | | SCA params |
| exemption | optional | string | The exemption that the transaction should take advantage of. Note that the requested exemption may not be accepted due to internal risk validations. Check SCA exemption values. |
| visa_merchant_id | required[5] | string(8) | VMID assigned by Visa if participating in Trusted merchant program. |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |
| **sender** | optional | | Funding Transaction Sender details |
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |
| **threeds_v2_params** | required* | | 3DSv2 async parameters. They must be submitted in order to use the 3DSv2 authentication protocol in asynchronous workflow |
| **threeds_method** | optional | | 3DS-Method related parameters for any callbacks and notifications. |

| | | | |
|---|---|---|---|
| callback_url | optional | url | Specific 3DS-Method callback URL after the 3DS-Method completes. The actual status will be provided via HTTP POST to that URL. For more information, go to 3DSv2 method params |
| **control** | required* | | General params for preferences in authentication flow and providing device interface information. |
| device_type | required* | string | Identifies the device channel of the consumer, **required** in the 3DSv2 authentication protocol. For more information, go to 3DSv2 control params |
| challenge_window_size | required* | string | Identifies the size of the challenge window for the consumer. For more information, go to 3DSv2 control params |
| challenge_indicator | optional | string | The value has weight and might impact the decision whether a challenge will be required for the transaction or not. If not provided, it will be interpreted as **no_preference**. For more information, go to 3DSv2 control params |
| **purchase** | optional | | Purchase related params providing with additional information regarding the order. |
| category | optional | string | Identifies the type of transaction being authenticated. This field is required in some markets.Accepted values are: **goods**, **service**, **check_acceptance**, **account_funding**, **quasi_cash**, **prepaid_activation**, **loan**. |
| **merchant_risk** | recommended | | Merchant risk assessment params. They are all optional, but recommended. |
| shipping_indicator | optional | string(16) | Indicator code that most accurately describes the shipping method for the cardholder specific transaction. If one or more items are included in the sale, use the Shipping Indicator code for the physical goods. If all digital goods, use the code that describes the most expensive item. Accepted values are: **same_as_billing**, **stored_address**, **verified_address**, **pick_up**, **digital_goods**, **travel**, **event_tickets**, **other**. |
| delivery_timeframe | optional | string(11) | Indicates the merchandise delivery timeframe. Accepted values are: **electronic**, **same_day**, **over_night**, **another_day**. |
| reorder_items_indicator | optional | string(10) | Indicates whether the cardholder is reordering previously purchased merchandise. Accepted values are: **first_time**, **reordered**. |
| pre_order_purchase_indicator | optional | string(21) | Indicates whether cardholder is placing an order for merchandise with a future-availability or release date. Accepted values are: **merchandise_available**, **future_availability**. |
| pre_order_date | optional | dd-mm-yyyy | For a pre-ordered purchase, the expected date that the merchandise will be available. |
| gift_card | optional | 'true' | Prepaid or gift card purchase. |
| gift_card_count | optional | integer | For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased. The value is limited to **99**. |
| **card_holder_account** | recommended | | Cardholder account additional information. They are all optional, but recommended, because they have a significant impact on approval rates |
| creation_date | optional | dd-mm-yyyy | Date that the cardholder opened the account with the 3DS Requester. |
| update_indicator | optional | string(19) | Length of time since the cardholder's account information with the 3DS Requestor was last changed. Includes Billing or Shipping address, new payment account, or new user(s) added. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| last_change_date | optional | dd-mm-yyyy | Date that the cardholder's account with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment accout, or new user(s) added. |
| password_change_indicator | optional | string(18) | Length of time since the cardholder account with the 3DS Requestor had a password change or account reset. Accepted values are: **no_change**, **during_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| password_change_date | optional | dd-mm-yyyy | Date that cardholder's account with the 3DS Requestor had a password change or account reset. |
| shipping_address_usage_indicator | optional | string(19) | Indicates when the shipping address used for this transaction was first used with the 3DS Requestor. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| shipping_address_date_first_used | optional | dd-mm-yyyy | Date when the shipping address used for this transaction was first used with the 3DS Requestor. |
| transactions_activity_last_24_hours | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours. |
| transactions_activity_previous_year | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year. |
| provision_attempts_last_24_hours | optional | integer | Number of Add Card attempts in the last 24 hours. |
| purchases_count_last_6_months | optional | integer | Number of purchases with this cardholder account during the previous six months. |
| suspicious_activity_indicator | optional | string(22) | Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account. Accepted values are: **no_suspicious_observed**, **suspicious_observed**. |
| registration_indicator | optional | string(19) | Indicates the length of time that the payment account was enrolled in the cardholder's account with the 3DS Requester. Accepted values are: **guest_checkout**, **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| registration_date | optional | dd-mm-yyyy | Date that the payment account was enrolled in the cardholder's account with the 3DS Requestor. |
| **browser** | required* | | For browser-based transactions. They are all *required* in case the **device_type** is set to **browser** |
| accept_header | required* | string(2048) | The exact content of the HTTP **ACCEPT** header as sent to the 3DS Requester from the Cardholder browser. Any other header different than the **ACCEPT** header will be rejected. Example: `application/json,text/plain,text/html,*/*`. |
| java_enabled | required* | boolean | Boolean that represents the ability of the cardholder browser to execute Java. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.javaEnabled`. |
| language | required* | string(8) | Value representing the browser language as defined in IETF BCP47. Note that only one browser language tag is about to be submitted as per the above **IETF BCP47**. Numeric chars are also allowed in the subtag and will represent the region.Example: `en-GB`, `zh-guoyu`, `fil-PH`, `gsw`, `es-419`, `de-1996`, etc. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.language`. |
| color_depth | required* | integer | Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from Cardholder browser using the `screen.colorDepth` property. The value as per EMVCo specs can be one of **1**, **4**, **8**, **15**, **16**, **24**, **32**, **48**. In case, an unsupported `color_depth` is determined, the nearest supported value that is less than the actual one needs to be submitted. For example, if the obtained value is **30**, which is not supported as per EMVCo specs, **24** has to be submitted. |
| screen_height | required* | integer | Total height of the Cardholder's screen in pixels. Value is returned from the screen.height property. |
| screen_width | required* | integer | Total width of the Cardholder's screen in pixels. Value is returned from the screen.width property. |
| time_zone_offset | required* | string(5) | Time difference between UTC time and the Cardholder browser local time, in **minutes**. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. If **UTC -5** hours then submit `300` or `+300`, If **UTC +2** hours then `-120`. The value can be retrieved using Javascript `getTimezoneOffset()` method over **Date** object. |
| user_agent | required* | string(2048) | Exact content of the HTTP user-agent header. |
| **sdk** | required* | | For application-based transactions. They are all *required* in case the **device_type** is set to **application** |
| interface | required* | string(6) | SDK Interface types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. Accepted values are: **native**, **html**, **both**. |
| **ui_types** | required* | | Lists all UI types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. |
| ui_type | required* | string(13) | UI type that the device of the consumer supports for displaying specific challenge interface. Accepted values are: **text**, **single_select**, **multi_select**, **out_of_bag**, **other_html**. |
| application_id | required* | string(36) | Universally unique ID created upon all installations and updates of the 3DS Requestor APp on a Customer Device. This will be newly generated and stored by the 3DS SDK for each installation or update. The field is limited to 36 characters and it shall have a canonical format as defined in IETF RFC 4122. |
| encrypted_data | required* | string(64000) | JWE Object as defined Section 6.2.2.1 containing data encrypted by the SDK for the DS to decrypt. The data will be present when sending to DS, but not present from DS to ACS. |
| ephemeral_public_key_pair | required* | string(256) | Public key component of the ephemeral key pair generated by the 3DS SDK and used to establish session keys between the 3DS SDK and ACS. In AReq, this data element is contained within the ACS Signed Content JWS Object. The field is limited to maximum 256 characters. |

| | | | |
|---|---|---|---|
| max_timeout | required* | integer | Indicates the maximum amount of time (in minutes) for all exchanges. The field shall have value greater or equals than 05. |
| reference_number | required* | string(32) | Identifies the vendor and version of the 3DS SDK that is integrated in a 3DS Requestor App, assigned by EMVCo when the 3DS SDK is approved. The field is limited to 32 characters. |
| **recurring** | optional | | Additional recurring details. |
| expiration_date | optional | dd-mm-yyyy | A future date indicating the end date for any further subsequent transactions. For more information, go to 3DSv2 recurring params |
| frequency | optional | integer | Indicates the minimum number of days between subsequent transactions. An empty value indicates the payment frequency is not set.For more information, go to 3DSv2 recurring params |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

**1** - Required if `mpi_params` is not present, the transaction will be handled asynchronously. Not required if configured on Terminal or Merchant level. Contact Tech Support for more details.

**2** - Required if transaction should be handled synchronous.

**3** - `eci` is always required if `mpi_params` is present.

`cavv` is not required for the 3D attempted only workflow, but it is strongly recommended in a combination with the Directory Server ID in order to be in the scope of the 3DSv2 authentication protocol.

**4** - `protocol_version` is required due to the only one 3DSv2 authentication protocol that is currently supported.

`directory_server_id` is mandatory when protocol_version is 2. May be omitted for scheme tokenized transactions.

**5** - `visa_merchant_id` is required when exemption value is `trusted_merchant`.

Frictionless / Challenge Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>init_recurring_sale3d</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <threeds_method_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method</threeds_method_url>
  <threeds_method_continue_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method/44177a21403427eb96664a6d7e5d5d48</threeds_method_continue_url>
  <timestamp>2025-02-26T16:01:35Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```

Challenge Without 3 Ds Method Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>init_recurring_sale3d</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <redirect_url>https://staging.gate.e-comprocessing.net/threeds/authentication/44177a21403427eb96664a6d7e5d5d48</redirect_url>
  <redirect_url_type>3ds_v2_challenge</redirect_url_type>
  <timestamp>2025-02-26T16:01:35Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |

| | | |
|---|---|---|
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| payment_account_reference | string(255) | Payment Account Reference value returned from the schemes. Links tokenized and PAN-based transactions. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where the consumer has to be redirected to complete the payment process unless a 3DSecure Method is required. This **redirect_url** will not be included in the response if a 3DS-Method submission is required. For more information, to go 3DSv2 authentication flows |
| redirect_url_type | string(64) | The type of the redirect URL in the 3DS scope. It will be present only for asynchronous 3D transactions when an interaction between consumer and issuer is required. This type identifies what kind of redirect url is returned, namely 3DSv2 Challenge. For more information, to go 3DSv2 authentication flows |
| threeds_method_url | url | 3DSecure Method URL. It will be present only then 3DS-Method is required for 3D transaction. A 3DS-Method submission inside an iframe is required to be submitted using HTTP POST. For more information, to go 3DSv2 authentication flows |
| threeds_method_continue_url | url | This is an API endpoint that accepts HTTP PUT & HTTP PATCH requests. It will be present when the **threeds_method_url** is included in the response. An HTTP PUT request must be submitted to that endpoint together with the proper signature to determine what the next step in the authentication is. For more information, to go 3DSv2 authentication flows |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to 'true' if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| scheme_transaction_link_id | string(22) | The transaction unique identifier returned from the schemes. Corresponds to ADDITIONAL SERVICE DATA (field 65, MasterCard Transaction Link Id [TLID]). |
| scheme_pan_indicator | string(1) | The account number indicator in scheme-tokenized transactions. See Account number indicator codes for details. |
| scheme_pan_tail | string(4) | The tail of the PAN number tail associated with the token in scheme-tokenized transactions. Returned in conjunction with `scheme_pan_indicator`. |
| scheme_max_settlement_date | string(10) | The final date an authorization could be captured which is returned from the card scheme. See Scheme Maximum Settlement Date for more details. |
| scheme_authentication_data_quality | boolean | The Authentication Data Quality Indicator returned by the card scheme is used to indicate whether a transaction meets the authentication data quality requirements. The value can be either `true` or `false`. |
| scheme_settlement_date | string(4) | MasterCard settlement date in MMDD format (e.g. 0227). Corresponds to NETWORK DATA (field 15). |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |
| **threeds** | | |
| eci | string(2) | See Electronic Commerce Indicator as returned from the MPI for details |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>init_recurring_sale3d</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <technical_message>expiration_year is invalid</technical_message>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:35Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| code | integer | Error code according to Error code table |

| | | |
|---|---|---|
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| **threeds** | | |
| **authentication** | | |
| status_reason_code | string(2) | See 3DS Authentication Status Reason Codes for details. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>init_recurring_sale3d</transaction_type>
  <status>declined</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>600</code>
  <technical_message>Cardholder not participating 3DS.</technical_message>
  <message>Transaction failed, please contact support!</message>
  <timestamp>2025-02-26T16:01:35Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <threeds>
    <eci>06</eci>
    <authentication>
      <status_reason_code>08</status_reason_code>
    </authentication>
  </threeds>
</payment_response>
```

**NOTIFICATION FOR ASYNCHRONOUS PAYMENTS**

When using asynchronous payments like 3D-Secure transactions, the transaction might depend on user input. Therefore, the outcome of the transaction is not available immediately after sending the request.

The gateway will send a notification to the merchant's server as soon as the payment has been completed (either approved or declined).

If the user cancels the payment (or the user has not completed the payment within the given time frame, e.g. 2 hours), the transaction will time out and a notification will be sent.

The notification will be sent as an HTTP POST to the `notification_url` specified in the transaction request XML. See Notifications for the HTTP POST-Data and format. The Notification will be pure HTTP or HTTPS-based, depending on the URL given by the merchant in the `notification_url`. In case it is a HTTPS-based notification, no SSL verification of the merchant SSL certificate will be performed. Until a notification echo is rendered by the merchant, there will be up to 10 notifications sent, each with a timeout of 15 minutes.

Note, the originated IP address of a notification will be one of the following per environment:

- Staging - **80.82.206.228**, **108.128.193.42**, **63.32.215.208**
- Production - **80.82.206.228**, **5.179.225.28**, **52.48.124.27**, **52.18.148.64**

Please, make sure IP addresses above are whitelisted respectively. The notifications are delivered to either port 80 (HTTP notification) or port 443 (HTTPS notification), so make sure the correct `notification_url` is submitted in the transaction request within the first place.

Also see 3-D secure workflow.

---

# Wallets

**NETELLER**

Neteller transactions are only synchronous. When the payment reaches a final state Genesis gateway sends notification to merchant on the configured url into its account.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
```

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>neteller</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <amount>100</amount>
  <currency>USD</currency>
  <customer_account>453501020503</customer_account>
  <account_password>908379</account_password>
  <customer_email>travis@example.com</customer_email>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **neteller** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| customer_account | required | string(100) | Customer email/id of neteller account |
| account_password | required | string(6) | Account secret password |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>neteller</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>neteller</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**PAYPAL**

PayPal trannsaction is a fast and easy way for buyers to pay with their PayPal account. It gives buyers all the transaction details at once, including order details, shipping options, insurance choices, and tax totals.

The following payment types are supported:

| Payment type | Description |
|---|---|
| authorize | Creates an order that should be captured later. |
| sale | Captures the created order immediately after the buyer confirms the payment. |
| express | Creates an Express Checkout PayPal payment. Express Checkout eliminates one of the major causes of checkout abandonment by giving buyers all the transaction details at once, including order details, shipping options, insurance choices, and tax totals. |

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>pay_pal</transaction_type>
   <transaction_id>1196432505475 01c79d8295</transaction_id>
   <payment_type>authorize</payment_type>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
```

```
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **pay_pal** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| payment_type | required | string | Use either **authorize** for Authorize or **sale** for Sale transactions |
| usage | optional | string(255) | Description of the transaction for later use. |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| document_id | required* | string(255) | Document ID value. |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>pay_pal</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ‘true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |

Error Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>pay_pal</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>expiration_year is invalid</technical_message>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```


**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**WEBMONEY**

WebMoney is a global settlement system and environment for online business activities. Over 28 million people from all over the world have joined the system. Although WebMoney is targeted mainly at clients in Russia and the Former Soviet Union, it is now used worldwide.

WebMoney transactions can be either asynchronous or synchronous, based on 'is_payout' flag. After a successful validation of transaction parameters, transaction status is set to pending_async, the user is redirected to WebMoney authentication page where he enters additional information to finish the payment. When the payment reaches a final state Genesis gateway sends notification to merchant on the configured url into its account. When the payment is checked as payout then the transaction is synchronous and transaction status is set immediately after the response.

Auth Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>webmoney</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <customer_phone>+1987987987987</customer_phone>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

Payout Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>webmoney</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <is_payout>true</is_payout>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <customer_phone>+1987987987987</customer_phone>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **webmoney** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| is_payout | required* | string | Value:true/false Flag for payout transaction |
| customer_account_id | required* | string(12) | Webmoney account ID (WMID). This field is required if is payout is set to "true" |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |

| | | | |
|---|---|---|---|
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Auth Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>webmoney</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

Payout Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>webmoney</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>webmoney</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

| | | |
|---|---|---|
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**WECHAT**

WeChat Pay solution offers merchants access to the over 300 million WeChat users that have linked payment accounts to their WeChat account. The solution works on desktop and mobile via a QR code generation platform.

Wechat payment transaction - after a successful validation of transaction parameters, transaction status is set to pending async and the consumer is redirected to page that contains the QR Code for scan. The consumer then opens the WeChat application on phone and scans the QR Code for payment confirmation. The gateway waits for an async notification from PaySec with the payment result of the consumer bank payment and updates the transaction status accordingly.

When the payment reaches a final state, the gateway sends a notification to the merchant on the configured notification URL for the merchant.

**Supported countries:** All countries are supported

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>wechat</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/a55ab44d242f</return_url>
   <amount>100</amount>
   <currency>USD</currency>
   <product_code>product-code</product_code>
   <product_num>1234</product_num>
   <product_desc>Product description</product_desc>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **wechat** |
| transaction_id | required | string(30) | Unique transaction id defined by merchant |
| usage | required | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| notification_url | required | url | URL at merchant where gateway sends outcome of transaction. |
| return_url | required | url | URL where consumer is sent to after payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details<br>Max amount in minor currency unit: 999999 |
| currency | required | string(3) | Currency code in ISO 4217 |
| product_code | optional | string(60) | Product code |
| product_num | optional | integer(10) | Product number |
| product_desc | optional | string(255) | Product description |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>wechat</transaction_type>
```

```
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>wechat</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Please check input data for errors!</message>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**ALIPAY**

Alipay is an oBeP-style alternative payment method that allows you to pay directly with your ebank account.

After initiating a transaction Alipay will redirect you to their page.

There you will see a picture of a QR code, which you will have to scan with your Alipay mobile application.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>alipay</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>CNY</currency>
   <customer_email>travis@example.com</customer_email>
```

```
    <billing_address>
      <first_name>Travis</first_name>
      <last_name>Pastrana</last_name>
      <address1>Muster Str. 12</address1>
      <zip_code>10178</zip_code>
      <city>Los Angeles</city>
      <neighborhood>Hollywood</neighborhood>
      <state>CA</state>
      <country>US</country>
    </billing_address>
  </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **alipay** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | optional | e-mail address | Must contain valid e-mail of customer |
| customer_phone | optional | string(32) | Must contain valid phone number of customer |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| **billing_address** | optional | | See Required vs Optional API params for details |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported currencies**

| Currency name | Currency code |
|---|---|
| China yen | CNY |
| Euro | EUR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>alipay</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>CNY</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>alipay</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>CNY</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### EZEEWALLET

eZeeWallet is a comprehensive digital wallet, connecting multiple payment methods simultaneously. It has the highest level of security, offering you a unique way to make secure and quick online payments. Consumers can upload funds to your account using P24, Trustly, Invoice (Sofort), iDeal, eps, NeoSurf and bank transfer and pay fast for goods and services with their available eZeeWallet balances.

> eZeeWallet and Genesis merchant accounts should be synced since authentication to eZeeWallet is done with Genesis merchant credentials

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>ezeewallet</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <source_wallet_id>john@example.com</source_wallet_id>
   <source_wallet_pwd>UDBydsDBrYWxAQA==\n</source_wallet_pwd>
   <amount>100</amount>
   <currency>USD</currency>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **ezeewallet** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| source_wallet_id | required | string(255) | Email address of consumer who owns the wallet |
| source_wallet_pwd | required | string(255) | Password of consumer who owns the wallet, in **Base64 encoded form** |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| notification_url | optional | url | URL at merchant where gateway sends outcome of transaction. |
| merchant_website | required | url | Website URL of the merchant initiating the wallet transfer |

`required* = conditionally required`

> return_success_url, return_failure_url and notification_url are used in case of top-up, i.e the customer does not have enough money to pay for the product

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<payment_response>
  <transaction_type>ezeewallet</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Pending Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ezeewallet</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:36Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Pending Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ezeewallet</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>930</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### EWALLET

eWallet transaction that handles different e-wallet providers

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>e_wallet</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_type>Free Charge</payment_type>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>50000</amount>
   <currency>INR</currency>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **e_wallet** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| payment_type | required | string | eWallet provider name that can be one of **Airtel Money**, **Amazon pay**, **Free Charge**, **Jio Money**, **Ola Money**, **Paytm**, **Payzapp**, **PhonePe**.Note, the list with the supported providers might be restricted based on the gateway configuration and currency. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |

| | | | |
|---|---|---|---|
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported currencies**

| Currency name | Currency code |
|---|---|
| Indian rupee | INR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>e_wallet</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>50000</amount>
  <currency>INR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>e_wallet</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <technical_message>amount is missing</technical_message>
  <message>Please check input data for errors!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>50000</amount>
  <currency>INR</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

# Vouchers

**CASHU**

CashU transactions are only asynchronous. After a successful validation of transaction parameters, transaction status is set to pending async, the user is redirected to CashU authentication page where he enters additional information to finish the payment. When the payment reaches a final state Genesis gateway sends notification to merchant on the configured url into its account.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>cashu</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>EUR</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **cashu** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries**

| Country name | Country code |
|---|---|
| Algeria | DZ |
| Bahrain | BH |
| Egypt | EG |
| Gambia | GM |

| | |
|---|---|
| Ghana | GH |
| India | IN |
| Iran | IR |
| Iraq | IQ |
| Israel | IL |
| Jordan | JO |
| Kenya | KE |
| Korea | KP |
| Kuwait | KW |
| Lebanon | LB |
| Libya | LY |
| Malaysia | MY |
| Mauritania | MR |
| Morocco | MA |
| Nigeria | NG |
| Oman | OM |
| Pakistan | PK |
| Palestine | PS |
| Qatar | QA |
| Saudi Arabia | SA |
| Sierra Leone | SL |
| Sudan | SD |
| Syria | SY |
| Tanzania | TZ |
| Tunisia | TN |
| Turkey | TR |
| United Arab Emirates | AE |
| United States | US |
| Yemen | YE |

**Supported currencies**

| Currency name | Currency code |
|---|---|
| Algerian Dinar | DZD |
| AmericanDollar | USD |
| EgyptianPound | EGP |
| Euro | EUR |
| JordanianDinar | JOD |
| LebanesePound | LBP |
| MoroccanDirham | MAD |
| Qatar Riyal | QAR |
| Saudi Riyal | SAR |
| Turkish Lira | TRY |
| UAE Dirham | AED |

Successful Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>cashu</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

| | | |
|---|---|---|
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>cashu</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**NEOSURF**

Neosurf is a prepaid card (voucher) that is used for online shopping. The card is available in over 100,000 stores worldwide, where customers can buy the prepaid vouchers, denominated up to EUR 250.00 or its equivalent in other currencies. This transaction is synchronous.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>neosurf</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>EUR</currency>
   <voucher_number>Vo23</voucher_number>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>23, Doestreet</address1>
     <zip_code>11923</zip_code>
     <city>New York City</city>
     <neighborhood>Manhattan</neighborhood>
     <country>DE</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **neosurf** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |

| | | | |
|---|---|---|---|
| voucher_number | required* | string(10) | Voucher number. Alphanumeric maximum 10 characters |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| return_success_url | required* | url | URL where customer is sent to after successful payment. It might be required depending on the gateway, it is strongly recommended for the URL to be set in order to avoid any invalidated transactions. |
| return_failure_url | required* | url | URL where customer is sent to after unsuccessful payment. It might be required depending on the gateway, it is strongly recommended for the URL to be set in order to avoid any invalidated transactions. |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries**

| Country Name | Country code |
|---|---|
| Austria | AT |
| Algeria | DZ |
| Australia | AU |
| Burundi | BI |
| Burkina Faso | BF |
| Benin | BJ |
| Belgium | BE |
| Cape Verde | CV |
| Cyprus | CY |
| Canada | CA |
| Central African Republic | CF |
| Chad | TD |
| Colombia | CO |
| Congo | CG |
| Cameroon | CM |
| Democratic Republic of Congo | CD |
| Denmark | DK |
| Equatorial Guinea | GQ |
| France | FR |
| Gambia | GM |
| Germany | DE |
| Gabon | GA |
| Guinea | GN |
| Ghana | GH |
| Guinea-Bissau | GW |
| Hong Kong | HK |
| Ireland | IE |
| Italy | IT |
| Ivory Coast | CI |
| Kenya | KE |

| | |
|---|---|
| Luxembourg | LU |
| Malawi | MW |
| Mozambique | MZ |
| Morocco | MA |
| Mauritania | MR |
| Mali | ML |
| Niger | NE |
| Nigeria | NG |
| Netherlands | NL |
| New Zealand | NZ |
| Norway | NO |
| Poland | PL |
| Portugal | PT |
| Rwanda | RW |
| Russia | RU |
| Romania | RO |
| Sweden | SE |
| Spain | ES |
| Sierra Leone | SL |
| Senegal | SN |
| Sao Tome and Principe | ST |
| Switzerland | CH |
| Serbia | RS |
| Turkey | TR |
| Togo | TG |
| Tunisia | TN |
| United Kingdom | GB |
| United Republic of Tanzania | TZ |
| Uganda | UG |
| Zimbabwe | ZW |
| Zambia | ZM |

**Supported currencies**

| Currency name | Currency code |
|---|---|
| Australian dollar | AUD |
| Bulgarian lev | BGN |
| Brazilian real | BRL |
| Canadian dollar | CAD |
| Swiss franc | CHF |
| Chinese yuan | CNY |
| Czech koruna | CZK |
| Danish krone | DKK |
| Euro | EUR |
| Pound sterling | GBP |
| Hong Kong dollar | HKD |
| Croatian kuna | HRK |
| Hungarian forint | HUF |
| Indonesian rupiah | IDR |
| Israeli new shekel | ILS |
| Indian rupee | INR |
| Japanese yen | JPY |
| South Korean won | KRW |
| Mexican peso | MXN |
| Malaysian ringgit | MYR |
| Norwegian krone | NOK |
| New Zealand dollar | NZD |
| Philippine peso | PHP |
| Polish złoty | PLN |
| Romanian leu | RON |
| Russian ruble | RUB |

| | |
|---|---|
| Swedish kronor | SEK |
| Singapore dollar | SGD |
| Thai baht | THB |
| Turkish lira | TRY |
| United States dollar | USD |
| CFA franc BCEAO | XOF |
| South African rand | ZAR |

Successful Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>neosurf</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>neosurf</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**PAYSAFECARD**

Paysafecard transactions are only asynchronous. After a successful validation of transaction param- eters transaction status is set to pending async the user is redirected to Paysafecard authentication page where he enters additional information to finish the payment. When the payment reaches a final state Genesis gateway sends notification to merchant on the configured url into its account.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>paysafecard</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_id>3192481752123</customer_id>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **paysafecard** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_id | required | string(32) | A value that uniquely identifies the end user (customer) and is disconnected from any personal information, it should be the same for all transactions of the customer. |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country name | Country code |
|---|---|
| Australia | AU |
| Austria | AT |
| Belgium | BE |
| Bulgaria | BG |
| Canada | CA |

| | |
|---|---|
| Croatia | HR |
| Cyprus | CY |
| Czech Republic | CZ |
| Denmark | DK |
| Finland | FI |
| France | FR |
| Georgia | GE |
| Germany | DE |
| Gibraltar | GI |
| Greece | GR |
| Hungary | HU |
| Iceland | IS |
| Ireland | IE |
| Italy | IT |
| Kuwait | KW |
| Latvia | LV |
| Liechtenstein | LI |
| Lithuania | LT |
| Luxembourg | LU |
| Malta | MT |
| Mexico | MX |
| Moldova | MD |
| Montenegro | ME |
| Netherlands | NL |
| New Zealand | NZ |
| Norway | NO |
| Paraguay | PY |
| Peru | PE |
| Poland | PL |
| Portugal | PT |
| Romania | RO |
| Saudi Arabia | SA |
| Slovakia | SK |
| Slovenia | SI |
| Spain | ES |
| Sweden | SE |
| Switzerland | CH |
| Turkey | TR |
| United Arab Emirates | AE |
| United Kingdom | GB |
| United States of America | US |
| Uruguay | UY |

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>paysafecard</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

| | | |
|---|---|---|
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>paysafecard</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

---

## Online Banking ePayments (oBeP)

Online Banking ePayments refer to payment methods that are used as an alternative to credit card payments allowing consumers to pay online with their bank account.

**BANCO DO BRASIL**

Banco do Brasil transanction will be soon deprecated. Please start using Online Banking transaction with BB bank code instead.

Banco do Brasil offers online bank transfer payment service.

Warning: We do not recommend using IFrames. This causes the scheme’s pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>banco_do_brasil</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rubble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Natal</city>
     <country>BR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
```

```
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **banco_do_brasil** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| BR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>banco_do_brasil</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |

| | | |
|---|---|---|
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>banco_do_brasil</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**BANCOMER**

Bancomer transanction will be soon deprecated. Please start using Online Banking transaction with BN bank code instead.

Bancomer offers two options for payments in Mexico, cash payment and bank transfer.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>bancomer</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rubble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Mexico City</city>
     <country>MX</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **bancomer** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |

| | | | |
|---|---|---|---|
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| MX |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bancomer</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bancomer</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**BRADESCO**

Bradesco transanction will be soon deprecated. Please start using Online Banking transaction with BR bank code instead.

Bradesco is a payment service in Brazil

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>bradesco</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <return_pending_url>http://www.example.com/pending</return_pending_url>
  <amount>100</amount>
  <currency>USD</currency>
  <consumer_reference>barney_rubble</consumer_reference>
  <national_id>8812128812</national_id>
  <birth_date>30-12-1992</birth_date>
  <customer_email>travis@example.com</customer_email>
  <billing_address>
    <first_name>Barney</first_name>
    <last_name>Rubble</last_name>
    <address1>14, Nerazdelni str</address1>
    <zip_code>1407</zip_code>
    <city>Rio de Janeiro</city>
    <country>BR</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **bradesco** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |

| | | | |
|---|---|---|---|
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| BR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bradesco</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bradesco</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
```

```
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**EPS**

EPS is the main bank transfer payment method in Austria. Every transaction is guaranteed via the scheme.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>eps</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>EUR</currency>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Vienna</city>
     <country>AT</country>
   </billing_address>
 </payment_transaction>'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>ppro</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_type>eps</payment_type>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>EUR</currency>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Vienna</city>
     <country>AT</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | **ppro** or **eps**. Contact tech support at tech-support@e-comprocessing.com for more details. |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| payment_type | required[1] | eps | EPS. Contact tech support for more details. |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |

| | | | |
|---|---|---|---|
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

`1 - payment_type must be submitted only when the transaction type is set to ppro`

**Supported countries and countries**

| Currency code | Country code |
|---|---|
| EUR | AT |

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ppro</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ppro</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
```

```
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**IDEBIT**

iDebit connects consumers to their online banking directly from checkout, enabling secure, real-time payments without a credit card. Using iDebit allows consumers to transfer funds to merchants without revealing their personal banking information.

The main difference between iDebit and InstaDebit is that InstaDebit uses eCheck, iDebit uses online bank transfer.

iDebit transactions have payins and payouts. The payin is asynchronous, while the payout is synchronous.

iDebit Payin transaction - after a successful validation of transaction parameters, transaction status is set to pending async and the consumer is redirected to the iDebit consumers page. The consumer then carries out the specified transaction details and finalizes the transaction. The gateway waits for an async notification from iDebit with the payment result of the consumer bank payment executed on the iDebit consumers page, and updates the transaction status accordingly.

When the payment reaches a final state, the gateway sends a notification to the merchant on the configured notification URL for the merchant. iDebit Payout transaction - the transaction is synchronous and transaction status is set immediately after the response.

**Supported countries**

| Country name | Country code |
|---|---|
| Canada | CA |

iDebit is available only for Canadian merchants and consumers.

**PAYIN**

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>idebit_payin</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <customer_account_id>1534537</customer_account_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/a55ab44d242f</return_url>
   <amount>100</amount>
   <currency>CAD</currency>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>46 Donora Dr</address1>
     <zip_code>M4B1B3</zip_code>
     <city>Toronto</city>
     <state>ON</state>
     <country>CA</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **idebit_payin** |
| transaction_id | required | string(30) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| customer_account_id | required | string(20) | Unique consumer account ID |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| **billing_address** | required | | See Required vs Optional API params for details |

| | | | |
|---|---|---|---|
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>idebit_payin</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>CAD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>idebit_payin</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>320</code>
  <message>amount is missing!</message>
  <timestamp>2025-02-26T16:01:37Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>CAD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

| | | |
|---|---|---|
| sent_to_acquirer | string(255) | "true" or "false" |

**PAYOUT**

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>idebit_payout</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <reference_id>43672</reference_id>
   <amount>100</amount>
   <currency>CAD</currency>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **idebit_payout** |
| transaction_id | required | string(30) | Unique transaction id defined by merchant |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>idebit_payout</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <technical_message>Transaction successful!</technical_message>
   <message>Transaction successful!</message>
   <timestamp>2025-02-26T16:01:38Z</timestamp>
   <amount>100</amount>
   <currency>CAD</currency>
   <sent_to_acquirer>true</sent_to_acquirer>
 </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>idebit_payout</transaction_type>
   <status>error</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <code>110</code>
   <technical_message>amount is missing!</technical_message>
   <message>Please check input data for errors!</message>
   <timestamp>2025-02-26T16:01:38Z</timestamp>
   <amount>100</amount>
   <currency>CAD</currency>
   <sent_to_acquirer>false</sent_to_acquirer>
 </payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

| | | |
|---|---|---|
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**IDEAL**

iDeal is the most popular payment method in the Netherlands and is a real-time bank transfer system covering all major Dutch consumer banks.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>ideal</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_transaction>payment_type</payment_transaction>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>EUR</currency>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Amsterdam</city>
     <country>NL</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **ideal** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| bic | optional | string(11) | SWIFT/BIC code of the customer's bank. If BIC is not provided, the consumer is redirected to a bank selection page. Get BIC list |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| notification_url | optional | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported currencies and countries**

| Currency code | Country code |
|---|---|
| EUR | NL |

Successful Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ideal</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ideal</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**Available issuers and their associated BIC**

| Bank Name | BIC |
|---|---|
| Rabobank | RABONL2U |
| ABN AMRO | ABNANL2A |
| Van Lanschot Bankiers | FVLBNL22 |
| Triodos Bank | TRIONL2U |
| ING Bank | INGBNL2A |
| SNS Bank | SNSBNL2A |

| | |
|---|---|
| ASN | ASNBNL21 |
| RegioBank | RBRBNL21 |
| Knab | KNABNL2H |
| Bunq | BUNQNL2A |
| Handelsbanken | HANDNL2A |
| Revolut | REVOLT21 |

**INSTADEBIT**

InstaDebit connects consumers to their online banking directly from checkout, enabling secure, realtime payments without a credit card. Using InstaDebit allows consumers to transfer funds to merchants without revealing their personal banking information.

The main difference between iDebit and InstaDebit is that InstaDebit uses eCheck, iDebit uses online bank transfer.

InstaDebit transactions have payins and payouts. The payin is asynchronous, while the payout is synchronous.

InstaDebit Payin transaction - after a successful validation of transaction parameters, transaction status is set to pending async and the consumer is redirected to the InstaDebit consumers page. The consumer then carries out the specified transaction details and finalizes the transaction. The gateway waits for an async notification from InstaDebit with the payment result of the consumer bank payment executed on the InstaDebit consumers page, and updates the transaction status accordingly.

When the payment reaches a final state, the gateway sends a notification to the merchant on the configured notification URL for the merchant.

InstaDebit Payout transaction - the transaction is synchronous and transaction status is set immediately after the response.

**Supported countries**

| Country name | Country code |
|---|---|
| Canada | CA |

InstaDebit is available only for Canadian merchants and consumers.

**PAYIN**

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>insta_debit_payin</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <customer_account_id>118221674199</customer_account_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/a55ab44d242f</return_url>
   <amount>100</amount>
   <currency>CAD</currency>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>46 Donora Dr</address1>
     <zip_code>M4B1B3</zip_code>
     <city>Toronto</city>
     <state>ON</state>
     <country>CA</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **insta_debit_payin** |
| transaction_id | required | string(30) | Unique transaction id defined by merchant |
| customer_account_id | required | string(20) | Unique consumer account ID |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_url | required | url | URL where consumer is sent to after payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>insta_debit_payin</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>CAD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>insta_debit_payin</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <message>Please check input data for errors!</message>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>CAD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**PAYOUT**

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <reference_id>43672</reference_id>
  <amount>100</amount>
  <currency>CAD</currency>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **insta_debit_payout** |

| | | | |
|---|---|---|---|
| transaction_id | required | string(30) | Unique transaction id defined by merchant |
| reference_id | required | string(32) | unique id of approved InstaDebit Payin transaction. See InstaDebit Payin Response, unique id |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details<br>Max amount in minor currency unit: 999999 |
| currency | required | string(3) | Currency code in ISO 4217 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>insta_debit_payout</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <amount>100</amount>
  <currency>CAD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>insta_debit_payout</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>320</code>
  <technical_message>amount is missing!</technical_message>
  <message>Please check input data for errors!</message>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <amount>100</amount>
  <currency>CAD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(30) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**ITAU**

Itau transanction will be soon deprecated. Please start using Online Banking transaction with IT bank code instead.

Itau is a real-time online bank transfer method and a virtual card.

Warning: We do not recommend using IFrames. This causes the scheme’s pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>itau</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rubble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Rio de Janeiro</city>
     <country>BR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **itau** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| BR |

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>itau</transaction_type>
   <status>pending_async</status>
   <mode>live</mode>
```

```
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>itau</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**MULTIBANCO**

Multibanco allows Portuguese shoppers to do payments through the Internet by using virtual credit cards

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>multibanco</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <return_pending_url>http://www.example.com/pending</return_pending_url>
  <amount>100</amount>
  <currency>EUR</currency>
```

```
    <customer_email>travis@example.com</customer_email>
    <billing_address>
      <first_name>Barney</first_name>
      <last_name>Rubble</last_name>
      <address1>14, Nerazdelni str</address1>
      <zip_code>1407</zip_code>
      <city>Porto</city>
      <country>PT</country>
    </billing_address>
    <risk_params>
      <user_id>123456</user_id>
    </risk_params>
  </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **multibanco** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | optional | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country code |
|---|
| PT |

Successful Response

Response

```
  <?xml version="1.0" encoding="UTF-8"?>
  <payment_response>
    <transaction_type>multibanco</transaction_type>
    <status>pending_async</status>
    <mode>live</mode>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
    <technical_message>Transaction successful!</technical_message>
    <message>Transaction successful!</message>
    <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
    <timestamp>2025-02-26T16:01:38Z</timestamp>
    <descriptor>Descriptor one</descriptor>
    <amount>100</amount>
    <currency>EUR</currency>
    <sent_to_acquirer>true</sent_to_acquirer>
  </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

| | | |
|---|---|---|
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>multibanco</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**MYBANK**

MyBank is an overlay banking system for Italy and Spain.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>my_bank</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <return_pending_url>http://www.example.com/pending</return_pending_url>
  <amount>100</amount>
  <currency>EUR</currency>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Barney</first_name>
    <last_name>Rubble</last_name>
    <address1>14, Nerazdelni str</address1>
    <zip_code>1407</zip_code>
    <city>Rome</city>
    <country>IT</country>
  </billing_address>
</payment_transaction>'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>ppro</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <payment_type>mybank</payment_type>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
```

```
  <amount>100</amount>
  <currency>EUR</currency>
  <billing_address>
    <first_name>Barney</first_name>
    <last_name>Rubble</last_name>
    <address1>14, Nerazdelni str</address1>
    <zip_code>1407</zip_code>
    <city>Rome</city>
    <country>IT</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | **ppro** or **my_bank**. Contact tech support at tech-support@e-comprocessing.com for more details. |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| payment_type | required[1] | ppro | MyBank. Contact tech support at tech-support@e-comprocessing.com for more details. |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

`1 - payment_type must be submitted only when the transaction type is set to ppro`

**Supported currencies and countries**

| Currency code | Country code |
|---|---|
| EUR | IT |
| EUR | BE |
| EUR | PT |
| EUR | ES |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ppro</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:38Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |

| | | |
|---|---|---|
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ppro</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:39Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### ONLINE BANKING

Online Banking is an oBeP-style alternative payment method that allows you to pay directly with your ebank account. After initiating a transaction, the online banking will redirect you to their page. There you will find a list with available banks to finish the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>online_banking</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>50000</amount>
   <currency>CNY</currency>
   <customer_email>travis@example.com</customer_email>
   <bank_code>CITIC</bank_code>
   <consumer_reference>Consumer Reference</consumer_reference>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **online_banking** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |

| | | | |
|---|---|---|---|
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| bank_code | required | bank code | Must contain Bank code |
| document_id | required* | string(255) | Document ID value. |
| payment_type | required* | string | The payment type describes the type of online banking used to process the transaction. Must contain one of the allowed Payment types, but they may vary based on the specific setup. If omitted, transaction will be processed with **online_banking** payment_type if online_banking is a supported payment type. Otherwise, the transaction will be processed with the first available supported payment type. |
| virtual_payment_address | required* | string | Virtual Payment Address (VPA) of the customer, format: someone@bank |
| consumer_reference | required* | string | Consumer reference identifier of the customer. |
| user_category | required* | string | User category. If missing, 'default' will be used. |
| auth_code | required* | string | 6-digit code used to authenticate the consumer within BLIK. It is required only for that bank. |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported currencies**

| Currency name | Currency code |
|---|---|
| Australian Dollar | AUD |
| China yen | CNY |
| Chilean Peso | CLP |
| Colombian Peso | COP |
| Indian rupee | INR |
| Indonesian rupiah | IDR |
| Malaysian ringgit | MYR |
| New Zealand dollar | NZD |
| Paraguayan Guarani | PYG |
| Philippine peso | PHP |
| Polish zloty | PLN |
| Singapore dollar | SGD |
| Thai baht | THB |
| Uruguayan Peso | UYU |
| Vietnamese dong | VND |
| Mexican Peso | MXN |

**Payment Types**

| Payment Type Name | Payment Type Code |
|---|---|
| Online banking | online_banking |
| Quick payment | quick_payment |
| Qr payment | qr_payment |
| Netbanking | netbanking |
| Alipay QR | alipay_qr |
| Bancomer | bancomer |
| Scotiabank | scotiabank |
| SPEI | spei |

Please, contact tech-support@e-comprocessing.com to find out more about which payment type to use with your setup.

**Bank Codes**

Bank codes may vary based on the specific setup.

**For AUD currency:**

| Bank Name | Bank Code |
|---|---|
| PayID | PID |

**For BRL currency:**

| Bank Name | Bank Code |
|---|---|
| Caixa | CA |

**For CAD currency:**

| Bank Name | Bank Code |
|---|---|
| Interac Combined Pay-in | CPI |

**For CHF currency:**

| Bank Name | Bank Code |
|---|---|
| Post Finance | PF |

**For CLP currency:**

| Bank Name | Bank Code |
|---|---|
| Servipag | SP |

**For CNY currency**

| Bank Name | Bank Code |
|---|---|
| Agricultural Bank of China | ABC |
| Bank of Beijing | BOB |
| Bank of China | BOC |
| Bank of Communications | BOCO |
| China Construction Bank | CCB |
| Bank for economic construction | CCD |
| China Everbright Bank | CEB |
| Industrial Bank | CIB |
| China Merchants Bank | CMB |
| China Minsheng Bank | CMBC |
| China Citic Bank | CITIC |
| Industrial and Commercial Bank of China | ICBC |
| China Guangfa Bank | GDB |
| Huaxia Bank | HXB |
| Pingan Bank | PINGANBANK |
| China Postal Savings Bank | PSBC |
| China Union Pay | QUICKPAY |
| Shanghai Bank | SHB |
| Shengzhen Ping An Bank | SPABANK |
| Shanghai Pudong Development Bank | SPDB |
| Yinlian Bank | YLB |

**For EUR currency:**

| Bank Name | Bank Code |
|---|---|
| Post Finance | PF |
| Bancontact | BCT |
| TrueLayer | TRL |

**For GBP currency:**

| Bank Name | Bank Code |
|---|---|
| TrueLayer | TRL |

**For IDR currency:**

| Bank Name | Bank Code |
|---|---|
| Bank Central Asia | DK_BCA_IB |
| Bank Rakyat Indonesia | DK_BRI_IB |
| CIMB Clicks Indonesia | DK_CIMBCLICKS_IB |
| Danamon Bank | DK_DANAMON_IB |
| Permata Bank | DK_PERMATANET_IB |

**For INR currency:**

| Bank Name | Bank Code |
|---|---|

| | |
|---|---|
| Aditya Birla Idea Payments Bank | ABPB |
| Airtel Payments Bank | AIRP |
| Allahabad Bank | ALLA |
| Andhra Bank | ANDB |
| Bank of Baroda - Retail Banking | BARB_R |
| Bank of Bahrein and Kuwait | BBKM |
| Dena Bank | BKDN |
| Bank of India | BKID |
| Central Bank of India | CBIN |
| City Union Bank | CIUB |
| Canara Bank | CNRB |
| Corporation Bank | CORP |
| Cosmos Co-operative Bank | COSB |
| Catholic Syrian Bank | CSBK |
| Development Bank of Singapore | DBSS |
| DCB Bank | DCBL |
| Deutsche Bank | DEUT |
| Dhanlaxmi Bank | DLXB |
| Equitas Small Finance Bank | ESFB |
| Federal Bank | FDRL |
| HDFC Bank | HDFC |
| IDBI | IBKL |
| ICICI Bank | ICIC |
| IDFC FIRST Bank | IDFB |
| Indian Bank | IDIB |
| Indusind Bank | INDB |
| Indian Overseas Bank | IOBA |
| Jammu and Kashmir Bank | JAKA |
| Janata Sahakari Bank (Pune) | JSBP |
| Karnataka Bank | KARB |
| Kotak Mahindra Bank | KKBK |
| Karur Vysya Bank | KVBL |
| Lakshmi Vilas Bank - Corporate Banking | LAVB_C |
| Lakshmi Vilas Bank - Retail Banking | LAVB_R |
| Bank of Maharashtra | MAHB |
| NKGSB Co-operative Bank | NKGS |
| Oriental Bank of Commerce | ORBC |
| Punjab & Maharashtra Co-operative Bank | PMCB |
| Punjab & Sind Bank | PSIB |
| Punjab National Bank - Retail Banking | PUNB_R |
| RBL Bank | RATN |
| State Bank of Bikaner and Jaipur | SBBJ |
| State Bank of Hyderabad | SBHY |
| State Bank of India | SBIN |
| State Bank of Mysore | SBMY |
| State Bank of Travancore | SBTR |
| Standard Chartered Bank | SCBL |
| South Indian Bank | SIBL |
| Saraswat Co-operative Bank | SRCB |
| State Bank of Patiala | STBP |
| Shamrao Vithal Co-operative Bank | SVCB |
| Syndicate Bank | SYNB |
| Tamilnadu Mercantile Bank | TMBL |
| Tamilnadu State Apex Co-operative Bank | TNSC |
| Union Bank of India | UBIN |
| UCO Bank | UCBA |
| United Bank of India | UTBI |
| Axis Bank | UTIB |
| Vijaya Bank | VIJB |

| | |
|---|---|
| Yes Bank | YESB |

**For MXN currency:**

| Bank Name | Bank Code |
|---|---|
| Spei | SE |
| Banorte | BQ |

**For MYR currency:**

| Bank Name | Bank Code |
|---|---|
| Affin Bank | FPX_ABB |
| Alliance Bank | FPX_ABMB |
| Am Online | FPX_AMB |
| Bank Islam | FPX_BIMB |
| Bank Muamalat | FPX_BMMB |
| Bank Rakyat | FPX_BKRM |
| Bank Simpanan Nasional | FPX_BSN |
| CIMB Clicks Bank | FPX_CIMBCLICKS |
| HLB Connect | FPX_HLB |
| Kuwait Finance House | FPX_KFH |
| Maybank2u | FPX_MB2U |
| OCBC Bank | FPX_OCBC |
| PBeBank | FPX_PBB |
| RHB Now | FPX_RHB |
| Stand Chart Bank | FPX_SCB |
| UOB Bank | FPX_UOB |

**For NZD currency:**

| Bank Name | Bank Code |
|---|---|
| Latipay | POLI |

**For PEN currency:**

| Bank Name | Bank Code |
|---|---|
| BCP | BC |
| Interbank | IB |
| Pago Efectivo | EF |
| BBVA | BP |

**For PLN currency:**

| Bank Name | Bank Code |
|---|---|
| BLIK | BLK |

**For PYG currency:**

| Bank Name | Bank Code |
|---|---|
| PagoExpress | PE |

**For THB currency:**

| Bank Name | Bank Code |
|---|---|
| Bangkok Bank | BBL_IB_U |
| Kasikornbank PAYPLUS | KBANK_PAYPLUS |
| Bank of Ayudhya (Krungsri) | BAY_IB_U |
| Krung Thai Bank | KTB_IB_U |
| Siam Commercial Bank | SCB_IB_U |

**For USD currency:**

| Bank Name | Bank Code |
|---|---|
| Santander | SN |
| Itau | IT |
| Bradesco | BR |
| Banco do Brasil | BB |
| Webpay | WP |
| Bancomer | BN |
| PSE | PS |
| Banco de Occidente | BO |

**For UYU currency:**

| Bank Name | Bank Code |
|---|---|

| | |
|---|---|
| Abitab | AI |

**For PHP currency:**

| Bank Name | Bank Code |
|---|---|
| Dragonpay | DRAGONPAY |

**For SGD currency:**

| Bank Name | Bank Code |
|---|---|
| DBS | ENETS-D_DBS |
| UOB | ENETS-D_UOB |
| OCBC | ENETS-D_OCBC |
| SCB | ENETS-D_SCB |

**For VND currency:**

| Bank Name | Bank Code |
|---|---|
| VTC-Pay VPBank | VTCP_VPBANK |
| VTC-Pay ABBANK | VTCP_ABBANK |
| VTC-Pay ACB | VTCP_ACB |
| VTC-Pay Agribank | VTCP_AGRIBANK |
| VTC-Pay BACABANK | VTCP_BACABANK |
| VTC-Pay BIDV | VTCP_BIDV |
| VTC-Pay BVB | VTCP_BVB |
| VTC-Pay DongABank | VTCP_DONGABANK |
| VTC-Pay Eximbank | VTCP_EXIMBANK |
| VTC-Pay GPBank | VTCP_GPBANK |
| VTC-Pay HDBank | VTCP_HDBANK |
| VTC-Pay LienVietPostBank | VTCP_LVPB |
| VTC-Pay MB | VTCP_MB |
| VTC-Pay MaritimeBank | VTCP_MARITIMEBANK |
| VTC-Pay NamABank | VTCP_NAMABANK |
| VTC-Pay Navibank | VTCP_NAVIBANK |
| VTC-Pay Oceanbank | VTCP_OCEANBANK |
| VTC-Pay PGBank | VTCP_PGBANK |
| VTC-Pay PHUONGDONG | VTCP_PHUONGDONG |
| VTC-Pay SHB | VTCP_SHB |
| VTC-Pay Sacombank | VTCP_SACOMBANK |
| VTC-Pay SaigonBank | VTCP-SAIGONBANK |
| VTC-Pay SeaABank | VTCP_SEAABANK |
| VTC-Pay Techcombank | VTCP_TECHCOMBANK |
| VTC-Pay TienPhong Bank | VTCP_TIENPHONGBANK |
| VTC-Pay VIB | VTCP_VIB |
| VTC-Pay VietABank | VTCP_VIETABANK |
| VTC-Pay Vietcombank | VTCP_VIETCOMBANK |
| VTC-Pay Vietinbank | VTCP_VIETINBANK |

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>online_banking</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:39Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>50000</amount>
  <currency>CNY</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <bank_code>CITIC</bank_code>
  <payment_type>online_banking</payment_type>
</payment_response>
```


Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>online_banking</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
```

```
    <message>Transaction successful!</message>
    <timestamp>2025-02-26T16:01:39Z</timestamp>
    <descriptor>Descriptor one</descriptor>
    <amount>50000</amount>
    <currency>MXN</currency>
    <payment_response>sent_to_acquirer</payment_response>
    <bank_code>SE</bank_code>
    <payment_response>payment_type</payment_response>
    <account_details>
      <reference_number>1000210</reference_number>
      <account_number>64618032000000006</account_number>
    </account_details>
  </payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | The redirect url is to be included, but also depend on the bank code and might not be returned for SE bank code when it is processed through the provider for the direct integration. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| bank_code | bank code | The bank code used to process the transaction, see Bank codes. |
| payment_type | string | The payment type describes the type of online banking used to process the transaction, see Payment types. |
| **account_details** | | |
| reference_number | string(7) | The reference number of the transaction to be used for the payment. |
| account_number | string(18) | The account number that the payment is expected to be transferred/done to. |

Error Response

Response


```
  <?xml version="1.0" encoding="UTF-8"?>
  <payment_response>
    <transaction_type>online_banking</transaction_type>
    <status>error</status>
    <mode>live</mode>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
    <code>110</code>
    <technical_message>amount is missing</technical_message>
    <message>Please check input data for errors!</message>
    <timestamp>2025-02-26T16:01:39Z</timestamp>
    <descriptor>Descriptor one</descriptor>
    <amount>50000</amount>
    <currency>CNY</currency>
    <sent_to_acquirer>true</sent_to_acquirer>
    <bank_code>CITIC</bank_code>
    <payment_type>online_banking</payment_type>
  </payment_response>
```


**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| bank_code | bank code | The bank code used to process the transaction, see Bank codes. |
| payment_type | string | The payment type describes the type of online banking used to process the transaction, see Payment types. |

**P24**

P24 transactions are only asynchronous. After a successful validation of transaction parameters, transaction status is set to pending async and the user is redirected to the P24 payment page where he enters additional information to finish the payment. When the payment reaches a final state, Genesis gateway sends a notification to the merchant.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>p24</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <payment_transaction>bank_code</payment_transaction>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **p24** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| bank_code | optional | integer | Must be one of the supported Bank codes |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**BANK CODES**

Bank codes may vary depending on the gateway configuration, please contact tech-support@e-comprocessing.com for further clarifications.

**For EUR and PLN currencies:**

| Bank Name | Bank Code |
|---|---|
| BLIK - PSP | 154 |
| EuroBank | 94 |
| Przekaz tradycyjny | 178 |
| Przekaz/Przelew tradycyjny | 1000 |
| Płac̨e z IKO | 135 |
| Płac̨e z Orange | 146 |

| | |
|---|---|
| Raiffeisen Bank PBL | 102 |
| U˙zyj przedpłaty | 177 |
| mBank-mTransfer | 25 |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>p24</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:39Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>p24</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:39Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**PAYU**

PayU is a payment method for Czech Republic and Poland

Warning: We do not recommend using IFrames. This causes the scheme’s pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>payu</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>PLN</currency>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Krakov</city>
     <country>PL</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **payu** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | optional | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required | string(255) | Customer first name |
| last_name | required | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported currencies and countries:**

| Currency code | Country code |
|---|---|
| CZK | CZ |
| PLN | PL |

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>payu</transaction_type>
   <status>pending_async</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <technical_message>Transaction successful!</technical_message>
   <message>Transaction successful!</message>
   <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
   <timestamp>2025-02-26T16:01:39Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>PLN</currency>
   <sent_to_acquirer>true</sent_to_acquirer>
 </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>payu</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:39Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>PLN</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**POLI**

POLi payment is Australia’s most popular online banking. The payment method is available within Australia and New Zealand. POLi transactions are asynchronous. After successful validation of transaction parameters, transaction status is set to pending async, the user is redirected to POLi authentication page where he enters additional information to finish the payment. When the payment reaches a final state Genesis gateway sends a notification to the merchant on the URL sent in the request or the URL configured in their account.

Note, in some rare cases, the POLi system might not be able to confirm whether the user’s payment is successful. In this case, the funds may have been transferred but the user will not be displayed a receipt from POLi or the merchant. This status can arise due to a bank issue or a local issue to the user.

**Therefore, POLi payments recommendation is for merchants to ensure that when end-user clicks on "Return to merchant's website", they land on a page that displays a clear message asking them to check their bank account before processing another transaction to make sure funds have not left the account.**

This will reduce the chances of duplicate transactions.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>poli</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <amount>100</amount>
  <currency>AUD</currency>
  <billing_address>
    <first_name>Barney</first_name>
    <last_name>Rubble</last_name>
```

```
    <address1>14, Nerazdelni str</address1>
    <zip_code>1407</zip_code>
    <city>Sofia</city>
    <country>AU</country>
    <state>AC</state>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **poli** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported bank countries:**

| Country name | Country code |
|---|---|
| Australia | AU |
| New Zealand | NZ |

**Supported currencies:**

| Currency name | Currency code |
|---|---|
| Australian dollar | AUD |
| New Zealand dollar | NZD |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>poli</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:39Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>AUD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |

| | | |
|---|---|---|
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>poli</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:39Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>AUD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**PSE (PAGOS SEGUROS EN LINEA)**

PSE transanction will be soon deprecated. Please start using Online Banking transaction with PS bank code instead.

PSE (Pagos Seguros en Linea) is the preferred alternative payment solution in Colombia.The solution consists of an interface that offers the client the option to pay for their online purchases in cash, directing it to their online banking.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>pse</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <return_pending_url>http://www.example.com/pending</return_pending_url>
  <amount>100</amount>
  <currency>USD</currency>
  <consumer_reference>barney_rumble</consumer_reference>
  <national_id>8812128812</national_id>
  <birth_date>30-12-1992</birth_date>
  <billing_address>
    <first_name>Barney</first_name>
    <last_name>Rubble</last_name>
    <address1>14, Nerazdelni str</address1>
    <zip_code>1407</zip_code>
    <city>Bogota</city>
    <country>CO</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **pse** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| CO |

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>pse</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:39Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |

| | | |
|---|---|---|
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>pse</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:39Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**POST FINANCE**

PostFinance transanction will be soon deprecated. Please start using Online Banking transaction with PF bank code instead.

PostFinance is an online banking provider in Switzerland

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>post_finance</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>EUR</currency>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Graz</city>
     <country>AT</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **post_finance** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| **billing_address** | required | | See Required vs Optional API params for details |

| | | | |
|---|---|---|---|
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported currencies and countries:**

| Currency code | Country code |
|---|---|
| EUR | CH |
| CHF | CH |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>post_finance</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <post_finance_transaction_id>438</post_finance_transaction_id>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| post_finance_transaction_id | string(255) | The Post Finance transaction ID |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <payment_response>transaction_type</payment_response>
  <payment_response>status</payment_response>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <payment_response>code</payment_response>
  <payment_response>message</payment_response>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <payment_response>sent_to_acquirer</payment_response>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**RAPIPAGO**

RapiPago from Argentina is an offline payment method used for online purchases. Shoppers buy their goods and services online and pay offline at one of the 6,000+ RapiPago payment locations.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>rapi_pago</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Buenos Aires</city>
     <country>AR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **rapi_pago** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |

| | | | |
|---|---|---|---|
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| AR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>rapi_pago</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>rapi_pago</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |

| | | |
|---|---|---|
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**SAFETYPAY**

Safetypay is a real-time bank transfer system that operates in more than 10 different countries. Their main market is in Latin America.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>safetypay</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_type>safetypay</payment_type>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>USD</currency>
   <payment_transaction>consumer_reference</payment_transaction>
   <payment_transaction>national_id</payment_transaction>
   <payment_transaction>birth_date</payment_transaction>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Tampico</city>
     <country>MX</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>ppro</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_type>safetypay</payment_type>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <payment_transaction>consumer_reference</payment_transaction>
   <payment_transaction>national_id</payment_transaction>
   <payment_transaction>birth_date</payment_transaction>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Tampico</city>
     <country>MX</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | **ppro** or **safetypay**. Contact tech support at tech-support@e-comprocessing.com for more details. |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| payment_type | required[1] | safetypay | SafetyPay. Contact tech support for more details |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |

| | | | |
|---|---|---|---|
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

`1 - payment_type must be submitted only when the transaction type is set to ppro`

**Supported countries:**

| Country |
|---|
| AT |
| BE |
| BR |
| CL |
| CO |
| DE |
| EC |
| ES |
| MX |
| NL |
| PE |
| PR |

**Supported currencies:**

| Currency Code |
|---|
| EUR |
| USD |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>safetypay</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ppro</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ppro/safetypay</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**SANTANDER**

Santander transanction will be soon deprecated. Please start using Online Banking transaction with SN bank code instead.

Santander is an online bank transfer for ecommerce purchases. Consumers use their trusted home banking environment, merchants benefit from payment guarantee and swift settlement.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>santander</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <return_pending_url>http://www.example.com/pending</return_pending_url>
  <amount>100</amount>
  <currency>USD</currency>
  <consumer_reference>barney_rubble</consumer_reference>
  <national_id>8812128812</national_id>
  <birth_date>30-12-1992</birth_date>
  <customer_email>travis@example.com</customer_email>
  <billing_address>
    <first_name>Barney</first_name>
    <last_name>Rubble</last_name>
    <address1>14, Nerazdelni str</address1>
    <zip_code>1407</zip_code>
    <city>Rio de Janeiro</city>
    <country>BR</country>
  </billing_address>
  <risk_params>
```

```
    <user_id>123456</user_id>
  </risk_params>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **santander** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| AR |
| BR |
| MX |
| CL |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>santander</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |

| | | |
|---|---|---|
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>santander</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### SDD INIT RECURRING SALE

An SddInitRecurringSale transaction initializes a recurring payment and is equal to a normal SddSaleTransaction except that it can be referenced as "initial" transaction in a SddRecurringSale transaction.

Note that if an SddInitRecurringSale is fully refunded, the recurring series is stopped and no more SddRecurringSales can be performed for that recurring series.

If an SddInitRecurringSale is partially refunded, the recurring series can continue with more SddRecurringSales.

> Authorize transactions are also available as 3dsecure transactions

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>sdd_init_recurring_sale</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_pending_url>http://www.example.com/pending</return_pending_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <amount>100</amount>
  <currency>EUR</currency>
  <iban>DE09100100101234567891</iban>
  <bic>PBNKDEFFXXX</bic>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <country>DE</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sdd_init_recurring_sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |

| | | | |
|---|---|---|---|
| return_success_url | required* | url | URL where customer is sent to after successful payment |
| return_pending_url | required* | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| return_failure_url | required* | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| iban | required | string(34) | Customer’s IBAN number |
| bic | optional | string(11) | SWIFT/BIC code of the customer’s bank |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required | string(255) | Customer first name |
| last_name | required | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country Name | Country Code |
|---|---|
| Austria | AT |
| Belgium | BE |
| Cyprus | CY |
| Estonia | EE |
| Finland | FI |
| France | FR |
| Germany | DE |
| Greece | GR |
| Ireland | IE |
| Italy | IT |
| Latvia | LV |
| Lithuania | LT |
| Luxembourg | LU |
| Malta | MT |
| Monaco | MC |
| Netherlands | NL |
| Portugal | PT |
| Slovakia | SK |
| San Marino | SM |
| Slovenia | SI |
| Spain | ES |

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sdd_init_recurring_sale</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
```

```
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sdd_init_recurring_sale</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Please check input data for errors!</message>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**SDD RECURRING SALE**

A SddRecurringSale transaction is a "repeated" transaction which follows and references a SddInitRecurringSale transaction.

The bank account data is omitted.

Note that SddRecurringSales can be partially or fully refunded if configuration allows it, and this will not stop the sdd recurring series.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sdd_recurring_sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>EUR</currency>
   <reference_id>2ee4287e67971380ef7f97d5743bb523</reference_id>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sdd_recurring_sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sdd_recurring_sale</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sdd_recurring_sale</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>320</code>
  <technical_message>amount is missing!</technical_message>
  <message>Please check input data for errors!</message>
  <timestamp>2025-02-26T16:01:40Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### SDD REFUND

**SddRefunds allow to return already billed amounts to customers.**

The amount can be fully refunded only, no partial refunds are allowed. Note that SDD refunds can only be done on former approved SDD transactions

Therefore, the reference_id for the corresponding transaction is mandatory.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sdd_refund</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <amount>100</amount>
   <currency>EUR</currency>
   <reference_id>2ee4287e67971380ef7f97d5743bb523</reference_id>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sdd_refund** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required | string(255) | Description of the transaction for later use. |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>sdd_refund</transaction_type>
   <status>pending_async</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <technical_message>Transaction successful!</technical_message>
   <message>Transaction successful!</message>
   <timestamp>2025-02-26T16:01:41Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>EUR</currency>
   <sent_to_acquirer>true</sent_to_acquirer>
 </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>sdd_refund</transaction_type>
   <status>error</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <code>320</code>
   <technical_message>amount is missing!</technical_message>
   <message>Please check input data for errors!</message>
   <timestamp>2025-02-26T16:01:41Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>EUR</currency>
   <sent_to_acquirer>false</sent_to_acquirer>
 </payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |

| | | |
|---|---|---|
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**SDD SALE**

The status of Sepa Direct Debit transactions is not available right after a transaction is made. Merchants receive the status of SDD transaction at 8:30 am (CET), 10:30 am (CET), 3:30 pm (CET) and 7:30 pm (CET). The merchant should have enabled notifications

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sdd_sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>EUR</currency>
   <iban>DE09100100101234567891</iban>
   <bic>PBNKDEFFXXX</bic>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <country>DE</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sdd_sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required* | url | URL where customer is sent to after successful payment |
| return_pending_url | required* | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| return_failure_url | required* | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| iban | required | string(34) | Customer's IBAN number |
| bic | optional | string(11) | SWIFT/BIC code of the customer's bank |
| company_name | optional | string(255) | Name of the company. |
| mandate_reference | optional | string(255) | Reference which contains the SEPAExpress paper mandate. |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required | string(255) | Customer first name |
| last_name | required | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |

| | | | |
|---|---|---|---|
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries: *

The supported countries are the same as SDD Init Recurring Sale.

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sdd_sale</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sdd_sale</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**SOFORT**

Sofort transactions are only asynchronous. After a successful validation of transaction parameters, transaction status is set to **pending_async**, the user is redirected to Sofort authentication page where he enters

additional information to finish the payment. When payment is still waiting for final state, it's state is set to **pending_hold**. As soon as the payment reaches a final state Genesis gateway sends notification to merchant on the configured url into its account.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sofort</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>EUR</currency>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Berlin</city>
     <country>DE</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sofort** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| notification_url | optional | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| iban | optional | string(24) | International bank account number of the customer |
| bic | optional | string(12) | Bank Identifier Code |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country name | Country code |
|---|---|
| Austria | AT |
| Belgium | BE |
| Germany | DE |

| | |
|---|---|
| Italy | IT |
| Netherlands | NL |
| Poland | PL |
| Spain | ES |
| Switzerland | CH |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sofort</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sofort</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### TRUSTLY SALE

Trustly is an oBeP-style alternative payment method that allows you to pay directly with your ebank account.

After initiating a transaction, Trustly will redirect the consumer to Trustly bank page. There the consumer will have to select his/her bank and log in with the regular access codes, choose the account and complete payment.

**Account_ID** parameter will be returned to the merchant notification url. **Account_ID** identifies each user's bank account once it is processed through Trustly system, it can be stored by the merchant and further used as a

reference on the Bank-Pay-out call.

When using your own hosted payment form, please follow Trustly requirements on services presentation and branding. Please contact your AM or our Technical team.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>trustly_sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_transaction>payment_type</payment_transaction>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_success_url_target>self</return_success_url_target>
   <amount>100</amount>
   <currency>EUR</currency>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Max</first_name>
     <last_name>Mustermann</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Berlin</city>
     <country>DE</country>
   </billing_address>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <payment_transaction>user_id</payment_transaction>
   <payment_transaction>account_id</payment_transaction>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **trustly_sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_success_url_target | optional | string(255) | URL target for successful payment in Trustly iFrame. Possible values: **self**, **parent**, **top**. |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| user_id | required* | string(255) | Unique user identifier defined by merchant in their own system. ID, username, hash or anything uniquely identifying the consumer requesting the deposit. Must be static per each consumer for any type of transaction where this consumer is involved (trustly_sale, bank pay_out, register_account, select account). |
| birth_date | optional | dd-mm-yyyy | Date of birth of the beneficiary, or organisational number for the organisation. |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| | | | Unique user Account identifier at Trustly system, which is used to process a Bank Pay-out call to the consumer, without reference to initial deposit transaction. You will |

| | | | |
|---|---|---|---|
| account_id | required | string(255) | receive this after Trustly_Sale and Select Account call on your notification URL. You will receive this as a response on Trustly Register Account option. |

`required* = conditionally required`

**Supported countries:**

| Country name | Country code |
|---|---|
| Austria | AT |
| Belgium | BE |
| Czech Republic | CZ |
| Denmark | DK |
| Estonia | EE |
| Finland | FI |
| Germany | DE |
| Latvia | LV |
| Lithuania | LT |
| Netherlands | NL |
| Norway | NO |
| Poland | PL |
| Slovakia | SK |
| Spain | ES |
| Sweden | SE |
| United Kingdom | GB |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>trustly_sale</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>trustly_sale</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |

| | | |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### UPI

UPI (Unified Payment Interface) transaction is an alternative payment method which allows users to transfer money between bank accounts.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>upi</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>50000</amount>
   <currency>INR</currency>
   <customer_email>travis@example.com</customer_email>
   <virtual_payment_address>someone@bank</virtual_payment_address>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **upi** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| document_id | required* | string(255) | Document ID value. |
| virtual_payment_address | required* | string(255) | Virtual Payment Address (VPA) of the customer, format: someone@bank |
| user_category | required* | string | User category. If missing, 'default' will be used. |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

> Virtual payment address is used and required for Unified Payment Interface (UPI) transactions.

**Supported currencies**

| Currency name | Currency code |
|---|---|

| | |
|---|---|
| Indian rupee | INR |

Successful Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>upi</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>50000</amount>
  <currency>INR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>upi</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <technical_message>amount is missing</technical_message>
  <message>Please check input data for errors!</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>50000</amount>
  <currency>INR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**WEBPAY**

Webpay transanction will be soon deprecated. Please start using Online Banking transaction with WP bank code instead.

Webpay is a Chilean real-time bank transfer method.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

This transaction type is refundable via Refund transaction.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>webpay</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>john_doe</consumer_reference>
   <national_id>8812128812</national_id>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Santiago</city>
     <country>CL</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **webpay** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries**

| Country Name | Country code |
|---|---|
| Chile | CL |

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <status>pending_async</status>
   <mode>live</mode>
```

```
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**DAVIVIENDA**

Davivienda is offering the Bill pay service which is a fast, easy and secure way to pay and manage your bills online to anyone, anytime in Colombia.

This transaction type is refundable via Refund transaction.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>davivienda</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_pending_url>http://www.example.com/pending</return_pending_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>barney@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
```

```
    <city>Bogota</city>
    <country>CO</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **davivienda** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | optional | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| CO |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>davivienda</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |

| | | |
|---|---|---|
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>davivienda</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

---

# Cash Payments

**CASH**

Cash payment methods allow customers to pay bills and online purchases in cash at convenient physical locations such as stores, banks, ATMs, even pharmacies in some countries. Usually, at checkout a voucher is generated with a barcode or another payment reference and the shopper can go to one of the supported shops/locations for the specific payment method and pay this voucher in cash.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>cash</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>50000</amount>
   <currency>MXN</currency>
   <customer_email>travis@example.com</customer_email>
   <payment_type>scotiabank</payment_type>
   <document_id>8812128812</document_id>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Monterrey</city>
     <country>MX</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **cash** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |

| | | | |
|---|---|---|---|
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| payment_type | required | string | The payment type describes desired payment method. Must contain one of the allowed Payment types, but they may vary based on the specific setup. |
| document_id | required | string(255) | Document ID value. |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported currencies**

| Currency name | Currency code |
|---|---|
| Mexican Peso | MXN |

**CASH PAYMENT TYPES**

| Payment Type Name | Payment Type Code |
|---|---|
| 7 Eleven | seven_eleven |
| Bancomer | bancomer |
| Farmacias del Dr. Ahorro | pharmacies_del_dr_ahorro |
| Farmacias Santa Maria | pharmacies_santa_maria |
| OXXO | oxxo |
| Scotiabank | scotiabank |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>cash</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>50000</amount>
  <currency>MXN</currency>
  <payment_type>scotiabank</payment_type>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| redirect_url | url | The redirect URL to the provider page for payment finalization. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| payment_type | string | The payment type describes the method used to process cash transaction. See Payment types. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
```

```
  <transaction_type>cash</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <code>110</code>
  <technical_message>amount is missing</technical_message>
  <message>Please check input data for errors!</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>50000</amount>
  <currency>MXN</currency>
  <payment_type>scotiabank</payment_type>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| payment_type | string | The payment type describes the method used to process cash transaction. See Payment types. |

**BALOTO**

Baloto is a cash payment option in Colombia. It allows the customers to receive a voucher at check-out. The voucher can then be paid in any of the Via Boleto offices in cash.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>baloto</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rubble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Bogota</city>
     <country>CO</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **baloto** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |

| | | | |
|---|---|---|---|
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| CO |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>baloto</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>baloto</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:41Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |

| | | |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**BANCO DE OCCIDENTE**

Banco de Occidente transanction will be soon deprecated. Please start using Online Banking transaction with BO bank code instead.

Banco de Occidente is a cash payment method for Colombia

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>banco_de_occidente</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Bogota</city>
     <country>CO</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **banco_de_occidente** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |

| | | | |
|---|---|---|---|
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| CO |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>banco_de_occidente</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>banco_de_occidente</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

| | | |
|---|---|---|
| sent_to_acquirer | string(255) | "true" or "false" |

**BOLETO**

Boleto is a payment service in Brazil

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>boleto</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rubble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Rio de Janeiro</city>
     <country>BR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **boleto** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| BR |

Successful Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>boleto</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <barcode>03395929800000015009008773800000000000270101</barcode>
  <ticket_expiry_date>04052022</ticket_expiry_date>
  <digitable_line>03399008707380000000900004301016489750000019000</digitable_line>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| barcode | string(44) | Barcode digit value acquired after transaction process |
| ticket_expiry_date | string(8) | Transaction expiry date in format %d%m%Y |
| digitable_line | string(47) | Digitable line value acquired after transaction process |

Error Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>boleto</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**EFECTY**

Efecty is a cash-based payment method.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>efecty</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>barney@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Bogota</city>
     <country>CO</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **efecty** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| CO |

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>efecty</transaction_type>
   <status>pending_async</status>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
   <technical_message>Transaction successful!</technical_message>
```

```
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>efecty</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**OXXO**

OXXO is the preferred payment method in Mexico. It is a cash payment via a barcode document thats accepted in more than 14,000 stores.

Warning: We do not recommend using IFrames. This causes the scheme’s pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>oxxo</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <amount>100</amount>
  <currency>USD</currency>
  <consumer_reference>barney_rumble</consumer_reference>
  <national_id>8812128812</national_id>
  <birth_date>30-12-1992</birth_date>
  <customer_email>barney@example.com</customer_email>
  <billing_address>
    <first_name>Barney</first_name>
```

```
    <last_name>Rubble</last_name>
    <address1>14, Nerazdelni str</address1>
    <zip_code>1407</zip_code>
    <city>Mexico City</city>
    <country>MX</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **oxxo** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| MX |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>oxxo</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |

| | | |
|---|---|---|
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>oxxo</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**PAGO FACIL**

Pago Facil is a cash-based payment used for online purchases.

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>pago_facil</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rumble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>barney@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Buenos Aires</city>
     <country>AR</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **pago_facil** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |

| | | | |
|---|---|---|---|
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| AR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>pago_facil</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>pago_facil</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**PIX**

Pix is a payment service created by the Central Bank of Brazil (BACEN), which represents a new way of receiving/sending money. Pix allows payments to be made instantly. The customer can pay bills, invoices, public utilities, transfer and receive credits in a facilitated manner, using only Pix keys (CPF/CNPJ).

Warning: We do not recommend using IFrames. This causes the scheme's pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>pix</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <return_pending_url>http://www.example.com/pending</return_pending_url>
  <amount>100</amount>
  <currency>USD</currency>
  <document_id>12345678909</document_id>
  <birth_date>30-12-1992</birth_date>
  <customer_email>travis@example.com</customer_email>
  <beneficiary>
    <name>ABC Corporation</name>
    <document_id>12345678000190</document_id>
    <pix_key>e89b1e34-18e0-40a7-9dd7-5a47d54a6b4f</pix_key>
  </beneficiary>
  <billing_address>
    <first_name>Barney</first_name>
    <last_name>Rubble</last_name>
    <address1>14, Nerazdelni str</address1>
    <zip_code>1407</zip_code>
    <city>Rio de Janeiro</city>
    <country>BR</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **pix** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required* | url | URL where customer is sent to after successful payment |
| return_failure_url | required* | url | URL where customer is sent to after unsuccessful payment |
| return_pending_url | required* | url | URL where customer is sent to when asynchronous payment is pending confirmation |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| document_id | required | string(255) | Document ID of the consumer. See Document ID Parameter for more details. |
| | | e-mail | |

| | | | |
|---|---|---|---|
| customer_email | required* | address | Must contain valid e-mail of customer |
| mothers_name | optional | string(255) | Mother's name of the customer. Submitting this parameter can increase the acceptance ratio for the payment method. |
| gender | optional | enum | 0 = Male, 1 = Female, 2 = Other. Submitting this parameter can increase the acceptance ratio for the payment method. |
| marital_status | optional | enum | 0 = NotMarried, 1 = Married, 2 = Divorced, 3 = Separate, 4 = Widower, 5 = Single, 6 = Other. Submitting this parameter can increase the acceptance ratio for the payment method. |
| sender_occupation | optional | string(255) | Occupation of the recepient. Submitting this parameter can increase the acceptance ratio for the payment method. |
| nationality | optional | string(255) | Nationality of the payer. Submitting this parameter can increase the acceptance ratio for the payment method. |
| country_of_origin | optional | string(255) | Country of origing of the payer. Two-letter iso codes. Submitting this parameter can increase the acceptance ratio for the payment method. |
| birth_city | optional | string(255) | City of birth of the payer. Submitting this parameter can increase the acceptance ratio for the payment method. |
| birth_state | optional | string(255) | Birth state of the payer. Submitting this parameter can increase the acceptance ratio for the payment method. |
| company_type | optional | enum | 0 = SA, 1 = LTDA, 2 = MEI, 3 = ME, 4 = EIRELI, 5 = Condominium, 6 = Closed SA, 7 = Simple EIRELI, 8 = Outros. Applicable for B2B payments only. Submitting this parameter can increase the acceptance ratio for the payment method. |
| company_activity | optional | string(255) | Type of company activity. Applicable for B2B payments only. Submitting this parameter can increase the acceptance ratio for the payment method. |
| birth_date | optional | string(10) | Birth date of the consumer |
| incorporation_date | optional | dd-mm-yyyy | Applicable for B2B payments only. Submitting this parameter can increase the acceptance ratio for the payment method. |
| **beneficiary** | optional | | The beneficiary parameters. Can only be submitted in special cases. Please contact Tech Support for more details. |
| name | optional | string(255) | The name of the beneficiary, either a Natural Person or Legal Person. |
| document_id | optional | string(14) | The Document ID of the beneficiary, such as CPF or CNPJ for Brazilian entities. |
| pix_key | optional | string(36) | The PIX key associated with the beneficiary. |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| BR |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>pix</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <emv_string>00020101021226870014br.gov.bcb.pix2565pix-h.santander.com.br/qr/v2/573119fe-8811-4612-9233-7252abc22ef4520400005303986580 2BR5925EMERCHANTPAY DO BRAZIL...6009SAO PAULO62070503***6304e675</emv_string>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |

| | | |
|---|---|---|
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| emv_string | string(255) | A string representation of the QR code. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>pix</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

**REDPAGOS**

Redpagos is a cash payment in Uruguay

Warning: We do not recommend using IFrames. This causes the scheme’s pages not to render correctly and not complete the payment.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>redpagos</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_reference>barney_rubble</consumer_reference>
   <national_id>8812128812</national_id>
   <birth_date>30-12-1992</birth_date>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Montevideo</city>
     <country>UY</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **redpagos** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |

| | | | |
|---|---|---|---|
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| consumer_reference | required | string(20) | Consumer reference is a unique consumer identifier |
| national_id | required | string(20) | National ID of the consumer. See Document ID Parameter for more details. |
| birth_date | optional | string(20) | Birth date of the customer |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

| Country |
|---|
| UY |

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>redpagos</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:42Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
```

```
    <transaction_type>redpagos</transaction_type>
    <status>error</status>
    <mode>live</mode>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
    <code>110</code>
    <message>Something went wrong, please contact support!</message>
    <timestamp>2025-02-26T16:01:42Z</timestamp>
    <descriptor>Descriptor one</descriptor>
    <amount>100</amount>
    <currency>USD</currency>
    <sent_to_acquirer>true</sent_to_acquirer>
  </payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

---

# Gift Cards

**INTERSOLVE**

**Intersolve transactions are made using gift card provided by Intersolve**

Using an intersolve transaction, the amount is immediately billed to the customer's gift card.

It can be reversed via a void transaction. Intersolve gift cards also support payout.

Use intersolve transactions if you are using gift cards provided by Intersolve.

> This transaction type supports Tokenization.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>intersolve</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_number>7000001163991388834</card_number>
   <cvv>944062</cvv>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **intersolve** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_number | required | string(19..21) | Gift card number |
| cvv | required* | 5 to 8 digits | Verification code of the gift card, requirement is based on terminal configuration |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| **billing_address** | required* | | See Required vs Optional API params for details |

| | | | |
|---|---|---|---|
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>intersolve</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |

| | | |
|---|---|---|
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>intersolve</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>Transaction_id is invalid!</technical_message>
  <message>Transaction_id is invalid!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

#### FASHIONCHEQUE

**Fashioncheque transactions are made using gift card provided by Fashioncheque**

Using a fashioncheque transaction, the amount is immediately billed to the customer's gift card.

It can be reversed via a void transaction on the same day of the transaction. They can also be refunded.

Use fashioncheque transactions, if you are using gift cards provided by Fashioncheque.

This transaction type supports Tokenization.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>fashioncheque</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_number>6046425117120757123</card_number>
  <cvv>121839</cvv>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **fashioncheque** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |

| | | | |
|---|---|---|---|
| currency | required | string(3) | Only USD and EUR |
| card_number | required | string(19..21) | Gift card number |
| cvv | required* | 5 to 8 digits | Verification code of the gift card, requirement is based on terminal configuration |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted |
| **billing_address** | required* | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>fashioncheque</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |

| | | |
|---|---|---|
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>fashioncheque</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>Transaction_id is invalid!</technical_message>
  <message>Transaction_id is invalid!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**TCS**

The container store **transactions are made using gift cards provided by TCS**.

The amount from a Container Store Transactions is immediately billed to the customer’s gift card.

It can be reversed via a void transaction.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>container_store</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_number>6046425117120757123</card_number>
  <cvv>121839</cvv>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **container_store** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Only USD and EUR |
| card_number | required | string(19..21) | Gift card number |

| | | | |
|---|---|---|---|
| cvv | required* | 5 to 8 digits | Verification code of the gift card, requirement is based on terminal configuration |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted |
| **billing_address** | required* | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>container_store</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |

| | | |
|---|---|---|
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>container_store</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>Transaction_id is invalid!</technical_message>
  <message>Transaction_id is invalid!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

#### SPLIT PAYMENTS

Split payments are performed on gift card transaction types when there isn't enough balance in the gift card. In order to use split payments you need to enable them on terminal. For more information please contact tech support.

**Split payments follow this workflow:**

- Split payment can be initiated only on gift card transaction.
- You can have maximum three payment series including the initial transaction.
- If the gift card does not have enough balance to perform the transaction, the whole available balance is taken from the gift card and new split payment is initiated.
- You can continue the split payment with another gift card
- You can finish the split payment with either gift card or credit card by submitting the 'unique id' of the initial transaction as 'reference id' in the request.
- Any failure during split payment causes rollback of all split payment series transactions.

Credit card transaction can only be last in split payment series and any series transactions must be submitted with the actual leftover amount.

**Example for initial split payment transaction:**

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>fashioncheque</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>5000</amount>
  <currency>EUR</currency>
  <card_number>6046425117120757123</card_number>
  <cvv>121839</cvv>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **fashioncheque** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |

| | | | |
|---|---|---|---|
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Only USD and EUR |
| card_number | required | string(19..21) | Gift card number |
| cvv | required* | 5 to 8 digits | Verification code of the gift card, requirement is based on terminal configuration |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted |
| **billing_address** | required* | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>fashioncheque</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>406bc1b340472db4dbbba4b749850234</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>2000</amount>
  <currency>EUR</currency>
  <split_payment>initiated</split_payment>
  <leftover_amount>3000</leftover_amount>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |

| | | |
|---|---|---|
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| split_payment | string | Split payment status for this transaction. Should be 'initiated'. |
| leftover_amount | integer | Leftover amount of transaction in minor currency unit, see Currency Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>fashioncheque</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>Transaction_id is invalid!</technical_message>
  <message>Transaction_id is invalid!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>5000</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**Example for continued split payment transaction:**

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>intersolve</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>406bc1b340472db4dbbba4b749850234</reference_id>
   <amount>3000</amount>
   <currency>EUR</currency>
   <card_number>7000001163991388834</card_number>
   <cvv>944062</cvv>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **intersolve** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |

| | | | |
|---|---|---|---|
| reference_id | required | string(32) | The ’unique id’ of the initial split payment transaction. |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_number | required | string(19..21) | Gift card number |
| cvv | required* | 5 to 8 digits | Verification code of the gift card, requirement is based on terminal configuration |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted |
| **billing_address** | required* | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>intersolve</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>2500</amount>
  <currency>EUR</currency>
  <split_payment>continued</split_payment>
  <leftover_amount>500</leftover_amount>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |

| | | |
|---|---|---|
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| split_payment | string | Split payment status for this transaction. Should be ‘continued’. |
| leftover_amount | integer | Leftover amount of transaction in minor currency unit, see Currency Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>intersolve</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>Transaction_id is invalid!</technical_message>
  <message>Transaction_id is invalid!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>3000</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**Example for finalized split payment transaction:**

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>intersolve</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>406bc1b340472db4dbbba4b749850234</reference_id>
   <amount>500</amount>
   <currency>EUR</currency>
   <card_number>7000001163991388834</card_number>
   <cvv>944062</cvv>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **intersolve** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |

| | | | |
|---|---|---|---|
| reference_id | required | string(32) | The ’unique id’ of the initial split payment transaction. |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_number | required | string(19..21) | Gift card number |
| cvv | required* | 5 to 8 digits | Verification code of the gift card, requirement is based on terminal configuration |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted |
| **billing_address** | required* | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>intersolve</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>500</amount>
  <currency>EUR</currency>
  <split_payment>finalized</split_payment>
  <leftover_amount>0</leftover_amount>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |

| | | |
|---|---|---|
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| split_payment | string | Split payment status for this transaction. Should be ’finalized’. |
| leftover_amount | integer | Leftover amount of transaction in minor currency unit, see Currency Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>intersolve</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>Transaction_id is invalid!</technical_message>
  <message>Transaction_id is invalid!</message>
  <timestamp>2025-02-26T16:01:43Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>500</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Split Payment can also be finalized using Sale or Sale3D

**Example for finalized Sale split payment transaction:**

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>sale</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <reference_id>406bc1b340472db4dbbba4b749850234</reference_id>
  <amount>500</amount>
  <currency>EUR</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4200000000000000</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required* | string(32) | The 'unique id' of the initial split payment transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <cvv_result_code>M</cvv_result_code>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>500</amount>
  <currency>EUR</currency>
  <split_payment>finalized</split_payment>
  <leftover_amount>0</leftover_amount>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| split_payment | string | Split payment status for this transaction. Should be ’finalized’. |
| leftover_amount | integer | Leftover amount of transaction in minor currency unit, see Currency Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| partial_approval | string(4) | Optional, set to ’true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Error Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <response_code>57</response_code>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>billing_address[zip_code] is invalid!</technical_message>
  <message>billing_address[zip_code] is invalid!</message>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>500</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```


**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

| | | |
|---|---|---|
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**Example for finalized Sale 3D split payment transaction:**

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sale3d** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | The 'unique id' of the initial split payment transaction |
| notification_url | required[1] | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required[1] | url | URL where customer is sent to after successful payment |
| return_failure_url | required[1] | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **mpi_params** | required[2] | | |
| cavv | required[3] | string(255) | Verification Id of the authentication. Please note this can be the CAVV for Visa Card or UCAF to identify MasterCard. |
| eci | required[3] | string(255) | See Electronic Commerce Indicator as returned from the MPI for details |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |

| | | | |
|---|---|---|---|
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

`1 - required if mpi_params is not present, the transaction will be handled asynchronously. Not required if configured on Terminal or Merchant level. Contact [tech-support@#{email_domain_name}](mailto:tech-support@#{email_domain_name}) for more details.`

`2 - required if transaction should be handled synchronous.`

`3 - eci is always required if mpi_params is present. cavv is not required for the 3D attempted only workflow, but it is strongly recommended in a combination with the Directory Server ID in order to be in the scope of the 3DSv2 authentication protocol.`

Successful Asynchronous Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>500</amount>
  <currency>EUR</currency>
  <split_payment>finalized</split_payment>
  <leftover_amount>0</leftover_amount>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ‘true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| split_payment | string | Split payment status for this transaction. Should be ‘finalized’. |
| leftover_amount | integer | Leftover amount of transaction in minor currency unit, see Currency Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <technical_message>expiration_year is invalid</technical_message>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>500</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**Voiding of split payments:**

- Voiding of any transaction in unfinished split payment (while still pending async) will cause rollback of all other transactions in split payment series. The current transaction will appear as voided while the other transaction will become declined.
- Voiding any transaction after the split payment has been completed will cause voiding only of the single transaction. In order to revert such split payment you need to manually void all transactions in it.
- The above points are valid also for refunding the transactions if the gift card gateway supports refund.

**Split Payment Timeouts:**

- All unfinished split payments will be automatically timed out after a period of time and all transactions in them will be rollbacked.
- If the split payment is finished with async 3D credit card transaction the split payment will be timed out according to the mpi timeout of the final transaction.

---

# Invoice Payment Methods

Alternative payments refer to payment methods that are used as an alternative to credit card payments.

Each alternative payment method has its own unique application, settlement process and currency support.

**INVOICE**

Klarna is a Swedish e-commerce company that provides payment services for online stores.

**With Invoice transactions, you can confirm that an order is successful.**

After settling the transaction (e.g. shipping the goods), you should use invoice capture transaction type to capture the amount.

Invoice transaction will automatically be cancelled after a certain time frame, most likely two weeks.

For a typical e-commerce application it is recommended to authorize the amount on incoming orders and capture it when shipping the goods.

If you choose not to serve the customer, consider to void the invoice to cancel the initial transaction.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>invoice</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_type>klarna</payment_type>
   <payment_method_category>pay_over_time</payment_method_category>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_cancel_url>http://www.example.com/cancel</return_cancel_url>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>60</amount>
   <currency>EUR</currency>
   <customer_phone>1987987987987</customer_phone>
   <customer_email>travis@example.com</customer_email>
   <customer_gender>male</customer_gender>
   <customer_birthdate>1990-03-20</customer_birthdate>
   <customer_reference_number>123</customer_reference_number>
   <order_tax_amount>0</order_tax_amount>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Berlin</city>
     <neighborhood>Lichtenberg</neighborhood>
     <state>Berlin</state>
     <country>DE</country>
   </billing_address>
   <shipping_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Berlin</city>
     <neighborhood>Lichtenberg</neighborhood>
     <state>Berlin</state>
     <country>DE</country>
   </shipping_address>
   <items>
```

```
    <item>
      <item_type>physical</item_type>
      <reference>19-402-USA</reference>
      <name>BatteryPowerPack</name>
      <quantity>1</quantity>
      <unit_price>60</unit_price>
      <tax_rate>0</tax_rate>
      <total_amount>60</total_amount>
      <total_discount_amount>0</total_discount_amount>
      <total_tax_amount>0</total_tax_amount>
      <image_url>https://example.com/image_url</image_url>
      <product_url>https://example.com/product_url</product_url>
      <quantity_unit>pcs</quantity_unit>
      <merchant_data>
        <marketplace_seller_info>Electronic gadgets</marketplace_seller_info>
      </merchant_data>
    </item>
  </items>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **invoice** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| payment_type | required | string | Payment provider type: **klarna** / **secure_invoice** |
| payment_method_category | required | string(255) | Payment method category: either **pay_over_time** or **pay_later** |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| customer_gender | optional | | Customer gender |
| customer_birthdate | required* | yyyy-mm-dd | Customer date of birth, required for Secure Invoice |
| customer_reference_number | required* | string(255) | Customer reference number, required for Secure Invoice |
| order_tax_amount | required | | Non-negative, minor units. The total tax amount of the order |
| **items** | required | | List with items |
| item_type | required | string(255) | Order line type. Possible values: Supported item types |
| quantity | required | integer | Non-negative. The item quantity |
| unit_price | required | integer | Minor units. Includes tax, excludes discount(max value: 100000000) |
| total_amount | required | integer | Includes tax and discount. Must match (quantity unit price) - total discount amount divided by quantity (max value: 100000000) |
| reference | optional | string(255) | Article number, SKU or similar |
| name | optional | string(255) | Descriptive item name |
| tax_rate | optional | integer | Non-negative. In percent, two implicit decimals. I.e 2500 = 25.00 percent |
| total_discount_amount | optional | integer | Non-negative minor units. Includes tax |
| total_tax_amount | optional | integer | Must be within 1 of total amount - total_amount * 10000 / (10000 + tax rate). Negative when type is discount |
| image_url | optional | url | URL to an image that can be later embedded in communications between Klarna and the customer. (max 1024 characters) |
| product_url | optional | url | URL to an image that can be later embedded in communications between Klarna and the customer. (max 1024 characters) |
| quantity_unit | optional | string(8) | Unit used to describe the quantity, e.g. kg, pcs... If defined has to be 1-8 characters |
| **product_identifiers** | optional | | List with product identifiers |

| | | | |
|---|---|---|---|
| brand | optional | string(255) | The product's brand name as generally recognized by consumers. If no brand is available for a product, do not supply any value |
| category_path | optional | string(255) | The product's category path as used in the merchant's webshop. Include the full and most detailed category and separate the segments with ' > ' |
| global_trade_item_number | optional | string(255) | The product's Global Trade Item Number (GTIN). Common types of GTIN are EAN, ISBN or UPC. Exclude dashes and spaces, where possible |
| manufacturer_part_number | optional | string(255) | The product's Manufacturer Part Number (MPN), which - together with the brand - uniquely identifies a product. Only submit MPNs assigned by a manufacturer and use the most specific MPN possible |
| **merchant_data** | optional | | List with merchant data |
| marketplace_seller_info | optional | string(255) | Information for merchant marketplace |

`required* = conditionally required`

Supported countries:

| Country | Country code |
|---|---|
| Austria | AT |
| Denmark | DK |
| Finland | FI |
| Germany | DE |
| Netherlands | NL |
| Norway | NO |
| Sweden | SE |

Supported item types:

| Item Types |
|---|
| physical |
| discount |
| shipping fee |
| sales tax |
| digital |
| gift card |
| store credit |
| surcharge |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>invoice</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>60</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>invoice</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <message>Something went wrong, please contact support!</message>
```

```
    <timestamp>2025-02-26T16:01:44Z</timestamp>
    <descriptor>Descriptor one</descriptor>
    <amount>60</amount>
    <currency>EUR</currency>
    <sent_to_acquirer>false</sent_to_acquirer>
  </payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |

#### INVOICE CAPTURE

**Invoice capture settles a Invoice transaction.**

Do this when you are shipping goods, for example. A Invoice capture can only be used after a Invoice on the same transaction.

Therefore, the `reference_id` of the Invoice transaction is mandatory.

> You can also use invoice capture for partial amount of the initial invoice authorize amount but invoice capture amount should be the same as the sum of items total amount. However, you cannot capture a higher amount than initially authorized.

**Transaction workflow:**

1. The merchant sends invoice transaction to the gateway.
2. The gateway replies to it. One of returned values is the unique id of the transaction.
3. The merchant sends invoice capture transaction. Its reference id is unique id of invoice response.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>invoice_capture</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_type>klarna</payment_type>
   <iban>DE09100100101234567891</iban>
   <account_holder>Ivan Ivanov</account_holder>
   <bank_transfer_remittance_slip>123123123</bank_transfer_remittance_slip>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
   <amount>60</amount>
   <currency>EUR</currency>
   <items>
     <item>
       <item_type>physical</item_type>
       <reference>19-402-USA</reference>
       <name>BatteryPowerPack</name>
       <quantity>1</quantity>
       <unit_price>60</unit_price>
       <tax_rate>0</tax_rate>
       <total_amount>60</total_amount>
       <total_discount_amount>0</total_discount_amount>
       <total_tax_amount>0</total_tax_amount>
       <image_url>https://example.com/image_url</image_url>
       <product_url>https://example.com/product_url</product_url>
       <quantity_unit>pcs</quantity_unit>
       <merchant_data>
         <marketplace_seller_info>Electronic gadgets</marketplace_seller_info>
       </merchant_data>
     </item>
   </items>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **invoice_capture** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| payment_type | required | string | Payment provider type: **klarna** / **secure_invoice** |
| account_holder | required* | string(255) | Account Holder, required for Secure Invoice in case of Direct Debit payment (payment_method_category: pay_over_time) |
| iban | required* | string(255) | IBAN, required for Secure Invoice in case of Direct Debit payment (payment_method_category: **pay_over_time**) |
| bank_transfer_remittance_slip | required* | string(255) | Bank Transfer Remittance Slip, required for Secure Invoice. Less then 16 symbols. |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |

| | | | |
|---|---|---|---|
| **items** | required | | List with items |
| item_type | required | string(255) | Order line type. Possible values: Supported item types |
| quantity | required | integer | Non-negative. The item quantity |
| unit_price | required | integer | Minor units. Includes tax, excludes discount(max value: 100000000) |
| total_amount | required | integer | Includes tax and discount. Must match (quantity unit price) - total discount amount divided by quantity (max value: 100000000) |
| reference | optional | string(255) | Article number, SKU or similar |
| name | optional | string(255) | Descriptive item name |
| tax_rate | optional | integer | Non-negative. In percent, two implicit decimals. I.e 2500 = 25.00 percent |
| total_discount_amount | optional | integer | Non-negative minor units. Includes tax |
| total_tax_amount | optional | integer | Must be within 1 of total amount - total_amount * 10000 / (10000 + tax rate). Negative when type is discount |
| image_url | optional | url | URL to an image that can be later embedded in communications between Klarna and the customer. (max 1024 characters) |
| product_url | optional | url | URL to an image that can be later embedded in communications between Klarna and the customer. (max 1024 characters) |
| quantity_unit | optional | string(8) | Unit used to describe the quantity, e.g. kg, pcs... If defined has to be 1-8 characters |
| **product_identifiers** | optional | | List with product identifiers |
| brand | optional | string(255) | The product's brand name as generally recognized by consumers. If no brand is available for a product, do not supply any value |
| category_path | optional | string(255) | The product's category path as used in the merchant's webshop. Include the full and most detailed category and separate the segments with ' > ' |
| global_trade_item_number | optional | string(255) | The product's Global Trade Item Number (GTIN). Common types of GTIN are EAN, ISBN or UPC. Exclude dashes and spaces, where possible |
| manufacturer_part_number | optional | string(255) | The product's Manufacturer Part Number (MPN), which - together with the brand - uniquely identifies a product. Only submit MPNs assigned by a manufacturer and use the most specific MPN possible |
| **merchant_data** | optional | | List with merchant data |
| marketplace_seller_info | optional | string(255) | Information for merchant marketplace |

`required* = conditionally required`

Supported countries:

| Country | Country code |
|---|---|
| Austria | AT |
| Denmark | DK |
| Finland | FI |
| Germany | DE |
| Netherlands | NL |
| Norway | NO |
| Sweden | SE |

Supported item types:

| Item Types |
|---|
| physical |
| discount |
| shipping_fee |
| sales_tax |
| digital |
| gift_card |
| store_credit |
| surcharge |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>invoice_capture</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>60</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

| | | |
|---|---|---|
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>invoice_capture</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>60</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| sent_to_acquirer | string(255) | "true" or "false" |

---

# Crypto

Alternative payment methods supporting digital cryptocurrencies.

**BITPAY SALE**

BitPay is a cryptocurrency payments provider supporting blockchain payments with Bitcoin (BTC) and BitcoinCash (BCH).

BitPay Sale is an asynchronous transaction type.

When this payment method is selected at checkout, the customer will be redirected to the BitPay system window including all the data for the payment: Bitcoin/BitcoinCash account, amount to be paid in cryptocurrency and the Fiat equivalent.

If the customer possesses a BitPay wallet or another BitPay compatible crypto wallet, the payment can be done from that window with one click, otherwise a QR CODE containing all the payment data can be scanned and used in any crypto wallet.

Then the customer has 15 minutes to fulfill the generated invoice.

If that timeframe is not met, the invoice will expire and the Merchant will be notified.

If the invoice is fulfilled in the timeframe, it needs to obtain 6 blockchain confirmations (1 hour) before it’s safe for the payment to be considered as completed.

At that point, the Merchant will be notified for the approved payment.

After the 6th confirmation, when the transaction is completed, the Merchant can process a refund if it’s needed.

**Supported countries**

| Country name | Country code |
|---|---|
| Afghanistan | AF |
| Aland Islands | AX |
| Albania | AL |
| American Samoa | AS |
| Andorra | AD |
| Angola | AO |
| Anguilla | AI |
| Antarctica | AQ |
| Antigua and Barbuda | AG |
| Argentina | AR |
| Armenia | AM |
| Aruba | AW |
| Australia | AU |

| | |
|---|---|
| Austria | AT |
| Azerbaijan | AZ |
| Bahamas | BS |
| Bahrain | BH |
| Barbados | BB |
| Belarus | BY |
| Belgium | BE |
| Belize | BZ |
| Benin | BJ |
| Bermuda | BM |
| Bhutan | BT |
| Bonaire, Sint Eustatius and Saba | BQ |
| Bosnia and Herzegovina | BA |
| Botswana | BW |
| Bouvet Island | BV |
| Brazil | BR |
| British Indian Ocean Territory | IO |
| Brunei Darussalam | BN |
| Bulgaria | BG |
| Burkina Faso | BF |
| Burundi | BI |
| Cameroon | CM |
| Canada | CA |
| Cape Verde | CV |
| Cayman Islands | KY |
| Central African Republic | CF |
| Chad | TD |
| Chile | CL |
| China | CN |
| Christmas Island | CX |
| Cocos (Keeling) Islands | CC |
| Colombia | CO |
| Comoros | KM |
| Congo | CG |
| Congo, the Democratic Republic of the | CD |
| Cook Islands | CK |
| Costa Rica | CR |
| Cote D'Ivoire | CI |
| Croatia | HR |
| Cuba | CU |
| Curacao | CW |
| Cyprus | CY |
| Czech Republic | CZ |
| Denmark | DK |
| Djibouti | DJ |
| Dominica | DM |
| Dominican Republic | DO |
| El Salvador | SV |
| Equatorial Guinea | GQ |
| Eritrea | ER |
| Estonia | EE |
| Ethiopia | ET |
| Falkland Islands (Malvinas) | FK |
| Faroe Islands | FO |
| Fiji | FJ |
| Finland | FI |
| France | FR |
| French Guiana | GF |
| French Polynesia | PF |

| | |
|---|---|
| French Southern Territories | TF |
| Gabon | GA |
| Gambia | GM |
| Georgia | GE |
| Germany | DE |
| Ghana | GH |
| Gibraltar | GI |
| Greece | GR |
| Greenland | GL |
| Grenada | GD |
| Guadeloupe | GP |
| Guam | GU |
| Guatemala | GT |
| Guernsey | GG |
| Guinea | GN |
| Guinea-Bissau | GW |
| Guyana | GY |
| Haiti | HT |
| Heard Island and McDonald Islands | HM |
| Holy See (Vatican City State) | VA |
| Honduras | HN |
| Hong Kong | HK |
| Hungary | HU |
| Iceland | IS |
| India | IN |
| Iran, Islamic Republic of | IR |
| Ireland | IE |
| Isle of Man | IM |
| Israel | IL |
| Italy | IT |
| Jamaica | JM |
| Japan | JP |
| Jersey | JE |
| Jordan | JO |
| Kazakhstan | KZ |
| Kenya | KE |
| Kiribati | KI |
| Korea, Democratic People's Republic of | KP |
| Korea, Republic of | KR |
| Kosovo, Republic of | XK |
| Kuwait | KW |
| Lao People's Democratic Republic | LA |
| Latvia | LV |
| Lebanon | LB |
| Lesotho | LS |
| Liberia | LR |
| Libyan Arab Jamahiriya | LY |
| Liechtenstein | LI |
| Lithuania | LT |
| Luxembourg | LU |
| Macao | MO |
| Madagascar | MG |
| Malawi | MW |
| Malaysia | MY |
| Maldives | MV |
| Mali | ML |
| Malta | MT |
| Marshall Islands | MH |
| Martinique | MQ |
| Mauritania | MR |

| | |
|---|---|
| Mauritius | MU |
| Mayotte | YT |
| Mexico | MX |
| Micronesia, Federated States of | FM |
| Moldova, Republic of | MD |
| Monaco | MC |
| Mongolia | MN |
| Montenegro | ME |
| Montserrat | MS |
| Mozambique | MZ |
| Myanmar | MM |
| Namibia | NA |
| Nauru | NR |
| Netherlands | NL |
| Netherlands Antilles | AN |
| New Caledonia | NC |
| New Zealand | NZ |
| Nicaragua | NI |
| Niger | NE |
| Nigeria | NG |
| Niue | NU |
| Norfolk Island | NF |
| Northern Mariana Islands | MP |
| Norway | NO |
| Oman | OM |
| Palau | PW |
| Panama | PA |
| Papua New Guinea | PG |
| Paraguay | PY |
| Peru | PE |
| Philippines | PH |
| Pitcairn | PN |
| Poland | PL |
| Portugal | PT |
| Puerto Rico | PR |
| Qatar | QA |
| Reunion | RE |
| Romania | RO |
| Russian Federation | RU |
| Rwanda | RW |
| Saint Barthélemy | BL |
| Saint Helena | SH |
| Saint Kitts and Nevis | KN |
| Saint Lucia | LC |
| Saint Martin French Part | MF |
| Saint Pierre and Miquelon | PM |
| Saint Vincent and the Grenadines | VC |
| Samoa | WS |
| San Marino | SM |
| Sao Tome and Principe | ST |
| Saudi Arabia | SA |
| Senegal | SN |
| Serbia | RS |
| Seychelles | SC |
| Sierra Leone | SL |
| Singapore | SG |
| Sint Maarten (Dutch part) | SX |
| Slovakia | SK |
| Slovenia | SI |

| | |
|---|---|
| Solomon Islands | SB |
| Somalia | SO |
| South Africa | ZA |
| South Georgia and the South Sandwich Islands | GS |
| South Sudan | SS |
| Spain | ES |
| Sri Lanka | LK |
| Sudan | SD |
| Suriname | SR |
| Svalbard and Jan Mayen | SJ |
| Swaziland | SZ |
| Sweden | SE |
| Switzerland | CH |
| Syrian Arab Republic | SY |
| Taiwan, Province of China | TW |
| Tajikistan | TJ |
| Tanzania, United Republic of | TZ |
| Thailand | TH |
| Timor-Leste | TL |
| Togo | TG |
| Tokelau | TK |
| Tonga | TO |
| Trinidad and Tobago | TT |
| Tunisia | TN |
| Turkmenistan | TM |
| Turks and Caicos Islands | TC |
| Tuvalu | TV |
| Uganda | UG |
| Ukraine | UA |
| United Arab Emirates | AE |
| United Kingdom | GB |
| United States | US |
| United States Minor Outlying Islands | UM |
| Uruguay | UY |
| Uzbekistan | UZ |
| Vanuatu | VU |
| Venezuela, Bolivarian Republic of | VE |
| Virgin Islands, British | VG |
| Virgin Islands, U.S. | VI |
| Wallis and Futuna | WF |
| Western Sahara | EH |
| Yemen | YE |
| Zambia | ZM |
| Zimbabwe | ZW |

**Supported currencies**

| Currency name | Currency code |
|---|---|
| Afghan Afghani | AFN |
| Albanian Lek | ALL |
| Angolan Kwanza | AOA |
| Argentine Peso | ARS |
| Armenian Dram | AMD |
| Aruban Florin | AWG |
| Australian Dollar | AUD |
| Azerbaijani Manat | AZN |
| Bahamian Dollar | BSD |
| Bahraini Dinar | BHD |
| Barbadian Dollar | BBD |
| Belarusian Ruble | BYN |
| Belize Dollar | BZD |
| Bermudan Dollar | BMD |

| | |
|---|---|
| Bermudan Dollar | BMD |
| Bhutanese Ngultrum | BTN |
| Bosnia-Herzegovina Convertible Mark | BAM |
| Botswanan Pula | BWP |
| Brazilian Real | BRL |
| British Pound Sterling | GBP |
| Brunei Dollar | BND |
| Bulgarian Lev | BGN |
| Burundian Franc | BIF |
| CFA Franc BCEAO | XOF |
| CFA Franc BEAC | XAF |
| CFP Franc | XPF |
| Canadian Dollar | CAD |
| Cape Verdean Escudo | CVE |
| Cayman Islands Dollar | KYD |
| Chilean Peso | CLP |
| Chinese Yuan | CNY |
| Colombian Peso | COP |
| Comorian Franc | KMF |
| Congolese Franc | CDF |
| Costa Rican Colón | CRC |
| Croatian Kuna | HRK |
| Cuba Pesos | CUP |
| Czech Republic Koruna | CZK |
| Danish Krone | DKK |
| Djiboutian Franc | DJF |
| Dominican Peso | DOP |
| East Caribbean Dollar | XCD |
| Eritrean Nakfa | ERN |
| Ethiopian Birr | ETB |
| Falkland Islands Pound | FKP |
| Fijian Dollar | FJD |
| Gambian Dalasi | GMD |
| Georgian Lari | GEL |
| Ghanaian Cedi | GHS |
| Gibraltar Pound | GIP |
| Guatemalan Quetzal | GTQ |
| Guinean Franc | GNF |
| Guyanaese Dollar | GYD |
| Haitian Gourde | HTG |
| Honduran Lempira | HNL |
| Hong Kong Dollar | HKD |
| Hungarian Forint | HUF |
| Icelandic Króna | ISK |
| Indian Rupee | INR |
| Iran, Rials | IRR |
| Israeli New Sheqel | ILS |
| Jamaican Dollar | JMD |
| Japanese Yen | JPY |
| Jordanian Dinar | JOD |
| Kazakhstani Tenge | KZT |
| Kenyan Shilling | KES |
| Korea (North), Won | KPW |
| Kuwaiti Dinar | KWD |
| Laotian Kip | LAK |
| Lebanese Pound | LBP |
| Lesotho Loti | LSL |
| Liberian Dollar | LRD |
| Libyan Dinar | LYD |
| Macanese Pataca | MOP |

| | |
|---|---|
| Malagasy Ariary | MGA |
| Malawian Kwacha | MWK |
| Malaysian Ringgit | MYR |
| Maldivian Rufiyaa | MVR |
| Mauritanian Ouguiya | MRU |
| Mauritian Rupee | MUR |
| Mexican Peso | MXN |
| Moldovan Leu | MDL |
| Mongolian Tugrik | MNT |
| Mozambican Metical | MZN |
| Myanma Kyat | MMK |
| Namibian Dollar | NAD |
| Netherlands Antillean Guilder | ANG |
| New Taiwan Dollar | TWD |
| New Zealand Dollar | NZD |
| Nicaraguan Córdoba | NIO |
| Nigerian Naira | NGN |
| Norwegian Krone | NOK |
| Omani Rial | OMR |
| Panamanian Balboa | PAB |
| Papua New Guinean Kina | PGK |
| Paraguayan Guarani | PYG |
| Peruvian Nuevo Sol | PEN |
| Philippine Peso | PHP |
| Polish Zloty | PLN |
| Qatari Rial | QAR |
| Romanian Leu | RON |
| Russian Ruble | RUB |
| Rwandan Franc | RWF |
| Saint Helena Pound | SHP |
| Salvadoran Colón | SVC |
| Samoan Tala | WST |
| Saudi Riyal | SAR |
| Serbian Dinar | RSD |
| Seychellois Rupee | SCR |
| Sierra Leonean Leone | SLL |
| Singapore Dollar | SGD |
| Solomon Islands Dollar | SBD |
| Somali Shilling | SOS |
| South African Rand | ZAR |
| South Korean Won | KRW |
| South Sudanese Pound | SSP |
| Sri Lankan Rupee | LKR |
| Sudan, Pounds | SDG |
| Surinamese Dollar | SRD |
| Swazi Lilangeni | SZL |
| Swedish Krona | SEK |
| Swiss Franc | CHF |
| Syria Pounds | SYP |
| São Tomé and Príncipe Dobra | STN |
| Tajikistani Somoni | TJS |
| Tanzanian Shilling | TZS |
| Thai Baht | THB |
| Tongan Pa‘anga | TOP |
| Trinidad and Tobago Dollar | TTD |
| Tunisian Dinar | TND |
| Turkmenistani Manat | TMT |
| Ugandan Shilling | UGX |
| Ukraine, Hryvnia | UAH |
| United Arab Emirates Dirham | AED |

| | |
|---|---|
| United Arab Emirates Dirham | AED |
| Uruguayan Peso | UYU |
| Uzbekistan Som | UZS |
| Vanuatu Vatu | VUV |
| Yemeni Rial | YER |
| Zambian Kwach | ZMW |

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>bitpay_sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/a55ab44d242f</return_url>
   <amount>3000</amount>
   <currency>EUR</currency>
   <customer_email>travis@example.com</customer_email>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Berlin</city>
     <country>DE</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **bitpay_sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | optional | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_url | required | url | URL where consumer is sent to after payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required* | | See Required vs Optional API params for details |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>bitpay_sale</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <technical_message>Transaction successful!</technical_message>
   <message>Transaction successful!</message>
   <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/649e1ff35c61</redirect_url>
   <timestamp>2025-02-26T16:01:44Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>3000</amount>
   <currency>EUR</currency>
 </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bitpay_sale</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <technical_message>Something went wrong, please contact support!</technical_message>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>3000</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

---

# Payouts

**BANK PAY-OUT**

Bank Pay-out option allows merchants to transfer funds directly to their consumers' bank account.

To process a bank pay-out via Trustly system, you need to have your customer's unique **Account_ID**. It is returned to your notification url during initial Trustly sale transaction. Alternatively, you may generate new customer's **Account_ID** via one the following steps: Trustly-register-account or Trustly-select-account

Once you've got your consumer's **Account_ID**, you may proceed to the Bank-Pay-out call.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>bank_payout</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>50000</amount>
   <currency>INR</currency>
   <customer_phone>+1987987987987</customer_phone>
   <customer_email>travis@example.com</customer_email>
   <bank_name>Netbanking</bank_name>
   <bank_code>321</bank_code>
   <bank_branch>HDFC0000001</bank_branch>
   <bank_account_number>1234123412341234</bank_account_number>
   <bank_account_name>Anurak Nghuen</bank_account_name>
   <id_card_number>123789456</id_card_number>
   <payer_bank_phone_number>01234567891</payer_bank_phone_number>
   <bank_account_type>C</bank_account_type>
   <bank_account_verification_digit>1</bank_account_verification_digit>
   <document_type>PASS</document_type>
```

```
    <payment_transaction>document_id</payment_transaction>
    <payment_type>bank_to_bank</payment_type>
    <billing_address>
      <first_name>Anurak</first_name>
      <last_name>Nghuen</last_name>
      <address1>Muster Str. 12</address1>
      <zip_code>10178</zip_code>
      <city>New Delhi</city>
      <state>New Delhi</state>
      <country>IN</country>
    </billing_address>
    <company_type>EIRELI</company_type>
    <company_activity>Mining</company_activity>
    <incorporation_date>2020-03-02</incorporation_date>
    <mothers_name>Maria</mothers_name>
    <pix_key>78690450041</pix_key>
  </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **bank_payout** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| notification_url | required | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| bank_name | optional | bank name | Name of the bank. If specified, it must be one of the supported Bank Names |
| bank_code | required | bank code | The bank code used to process the transaction. Must be one of the supported Bank codes. |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required[1] | string(32) | Must contain valid phone number of customer |
| bank_branch | required* | bank branch | Name of the Bank branch. |
| bank_account_name | required* | string(255) | Bank account name is required, for CNY currency and should be in Simplified Chinese. For other currency, must be in English Language. |
| bank_account_number | required* | bank account number | Bank account number of the customer. |
| bank_province | required* | bank province | Name of the province that the bank is located. |
| id_card_number | required* | id card number | ID card number. See Document ID Parameter for more details. |
| payer_bank_phone_number | required* | string(11) | Payer bank phone number |
| bank_account_type | required* | string(1) | The type of account. C: for Checking accounts S: for Savings accounts M: for Maestra accounts(Only Peru) P: for Payment accounts |
| bank_account_verification_digit | required* | string(1) | Verifier digit. Given by external provider, used to verify transaction. |
| document_type | required* | string(10) | ID card/document type |
| document_id | required* | string(255) | Document ID value. |
| account_id | required* | string(255) | Unique account identifier in Trustly's system. You will receive this after Select Account call and after Trustly Sale on the notification URL. |
| user_id | required* | string(255) | Unique user identifier defined by merchant in their own system. ID, username, hash or anything uniquely identifying the consumer requesting the deposit. Must be static per each consumer for any type of transaction where this consumer is involved (trustly_sale, bank pay_out, register_account, select account). |
| birth_date | required* | dd-mm-yyyy | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| payment_type | required* | string(12) | Bank payout subtype. Available values: bank_to_bank, pix, bsb, pay_id, bank_to_bank_b2b, pix_b2b, clabe, cellphone. |
| company_type | required* | string(255) | Company type of the customer. For Legal Person. |
| company_activity | required* | string(255) | Company activity of the customer. For Legal Person. |
| incorporation_date | required* | yyyy-mm-dd | The incoroporation date of the customer. For Legal Person. |
| mothers_name | required* | string(255) | Mother's name of the customer. |
| pix_key | required* | string(255) | PIX key of the customer. |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required | string(255) | Customer first name |
| last_name | required | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

required* = conditionally required

**1** - Only for Brazilian transactions allowed characters are as follows:

- 12345678901: Exactly 11 digits.
- +1234567890123: An optional plus sign followed by exactly 13 digits.

**Supported currencies**

| Currency name | Currency code |
|---|---|
| Argentine peso | ARS |
| Brazilian real | BRL |
| Chilean peso | CLP |
| China yen | CNY |
| Colombian peso | COP |
| Indonesian rupiah | IDR |
| Indian rupee | INR |
| Malaysian ringgit | MYR |
| Mexican peso | MXN |
| Peruvian sol | PEN |
| Thai baht | THB |
| Uruguayan peso | UYU |

**BANK NAMES**

Bank Names may vary based on the specific setup.

**For CAD currency:**

| Bank Name |
|---|
| Interac e-Transfer Outbound Pay-out |
| eCashout Pay-out |

**For CNY currency:**

| Bank Name |
|---|
| |
| |
| |

| |
|---|
| |
| |
| |

**For MYR currency:**

| Bank Name |
|---|
| 423 |
| CIMB Clicks Bank |
| Hong Leong Bank |
| May Bank |
| Public Bank |
| RHB Bank |

**For THB currency:**

| Bank Name |
|---|
| Bangkok Bank |
| Kasikorn Bank |
| Krungsri (Bank of Ayudhya Public Company Limited) |
| Krung Thai Bank |
| Siam Commercial Bank |
| UOBT |

**For IDR currency:**

| Bank Name |
|---|
| Bank Central Asia |
| Bank Rakyat Indonesia |
| Bank Negara Indonesia |
| BTN Bank |
| CIMB Clicks Indonesia |
| Danamon Bank |
| Mandiri Bank |
| Permata Bank |

**For INR currency:**

| Bank Name |
|---|
| ABHYUDAYA COOP BANK |
| THE ROYAL BANK OF SCOTLAND |
| ABU DHABI COMMERCIAL BANK |
| THE AKOLA DISTRICT CENTRAL COOPERATIVE BANK |
| AIRTEL PAYMENTS BANK LIMITED |
| AKOLA JANATA COMMERCIAL COOPERATIVE BANK |
| ALLAHABAD BANK |
| THE AHMEDABAD MERC COOP BANK |
| ANDHRA BANK |
| AUSTRALIA & NEW ZEALAND BANK |
| THE ANDHRA PRADESH STATE COOP BANK |
| ANDHRA PRAGATI GRAMEEN BANK |
| THE A.P. MAHESH CO-OP URBAN BANK |
| APNA SAHAKARI BANK LTD |
| ALMORA URBAN CO-OPERATIVE BANK LTD. |
| BASSEIN CATHOLIC CO-OP BANK |
| BANK OF BARODA |
| BARCLAYS BANK |
| BANK OF BAHREIN & KUWAIT |
| THE BHARAT COOPERATIVE BANK |
| BANK OF CEYLON |
| BANDHAN BANK LIMITED |
| DENA BANK |
| BANK OF INDIA |
| BHARATIYA MAHILA BANK LIMITED |
| B N PARIBAS BANK |
| BANK OF AMERICA |

| |
|---|
| BANK OF TOKYO-MITSUBISHI |
| CENTRAL BANK OF INDIA |
| CITIZEN CREDIT COOP BANK |
| JP MORGAN CHASE BANK |
| CITI BANK |
| CITY UNION BANK |
| CAPITAL LOCAL AREA BANK LTD. |
| CANARA BANK |
| CORPORATION BANK |
| THE COSMOS CO-OP. BANK |
| CREDIT SUISSE AG? |
| CREDIT AGRICOLE CORP N INVSMNT BK |
| CHHATRAPATI RAJARSHISHAHU COOP BANK |
| CATHOLIC SYRIAN BANK |
| COMMONWEALTH BK OF AUSTRALIA |
| CHINATRUST COMMERCIAL BANK |
| DEVELOPMENT BANK OF SINGAPORE |
| DEVELOPMENT CREDIT BANK |
| DEOGIRI NAGARI SAHAKARI BANK LTD. AURANGABAD |
| DEUTSCHE BANK |
| DICGC |
| THE DELHI STATE COOPERATIVE BANK LIMITED |
| DHANALAXMI BANK |
| DOMBIVLI NAGARI SAHAKARI BANK LTD |
| DOHA BANK QSC |
| EXPORT IMPORT BANK OF INDIA |
| EQUITAS SMALL FINANCE BANK LIMITED |
| THE FEDERAL BANK |
| FIRSTRAND BANK |
| THE GREATER BOMBAY CO-OP. BANK LTD |
| THE GADCHIROLI DISTRICT CENTRAL COOPERATIVE BANK LIMITED |
| GURGAON GRAMIN BANK LTD. |
| THE GUJARAT STATE CO-OPERATIVE BANK |
| THE HASTI COOP BANK LTD |
| HDFC BANK LTD. |
| HIMACHAL PRADESH STATE COOPERATIVE BANK LTD |
| HONG KONG & SHANGHAI BANK |
| Woori |
| PT BANK MAYBANK INDONESIA TBK |
| IDBI BANK |
| INDUSTRIAL BANK OF KOREA |
| INDUSTRIAL AND COMMERCIAL BANK OF CHINA LIMITED |
| ICICI BANK LTD. |
| IDFC BANK LIMITED |
| INDIAN BANK |
| IDUKKI DISTRICT CO OPERATIVE BANK LTD |
| INDUS-IND BANK |
| INDIAN OVERSEAS BANK |
| THE JAMMU & KASHMIR BANK |
| JANSEVA SHAHKARI BANK LTD. PUNE |
| JANASEVA SAHAKARI BANK BORIVLI LIMITED |
| JALGAON JANATA SAHAKARI |
| THE JALGAON PEOPELS COOPERATIVE BANK LIMITED |
| JANKALYAN SHAKARI BANK |
| JANATA SAHAKARI BANK LTD (PUNE) |
| THE KANGRA CENTRAL COOPERATIVE BANK |
| KALLAPPANNA AWADE ICH JANATA S |
| THE KANGRA COOPERATIVE BANK LTD |
| KARNATAKA BANK |
| KAPOLE BANK |

| |
|---|
| THE KALUPUR COMM COOP BANK |
| THE KALYAN JANATA SAHAKARI BANK |
| KOTAK MAHINDRA BANK |
| KERALA GRAMIN BANK |
| THE KURMANCHAL NAGAR SAHAKARI BANK LIMITED |
| THE KARNATAKA STATE COOP APEX BANK |
| KEB Hana Bank |
| THE KARAD URBAN COOP BANK LTD |
| KARUR VYSYA BANK |
| KARNATAKA GRAMIN VIKAS BANK |
| THE LAKSHMI VILAS BANK |
| BANK OF MAHARASHTRA |
| Maharashtra Gramin Bank |
| MAHANAGAR COOP BANK |
| MUMBAI DISTRICT CENTRAL CO-OP BANK |
| MIZUHO CORPORATE BANK LTD |
| Maharashtra State Cooperative Bank |
| MASHREQ BANK |
| THE MEHSANA URBAN COOPERATIVE BANK |
| THE MUNICIPAL CO OPERATIVE BANK LTD |
| NATIONAL AUSTRALIA BANK LIMITED |
| NATIONAL BANK OF ABU DHABI PJSC |
| NAGPUR NAGRIK (NNSB LTD*) |
| NEW INDIA CO-OPERATIVE BANK |
| NKGSB BANK |
| THE NASIK MERCHANTS CO-OP BANK LTD. |
| NORTH MALBAR GRAMIN BANK |
| NUTAN NAGARIK SAHAKARI BANK |
| THE BANK OF NOVA SCOTIA |
| THE NAINITAL BANK LTD |
| NAGAR URBAN CO OPERATIVE BANK |
| OMAN INTERNATIONAL BANK |
| ORIENTAL BANK OF COMMERCE |
| PARSIK JANATA SAHAKARI BANK |
| PRAGATHI KRISHNA GRAMIN BANK |
| PUNJAB AND MAHARASHTRA CO-OP BANK |
| PRIME CO OPERATIVE BANK LTD |
| PRATHAMA BANK |
| PUNJAB AND SIND BANK |
| THE PANDHARPUR URBAN CO OP. BANK LTD. PANDHARPUR |
| PUNJAB NATIONAL BANK |
| RABOBANK INTERNATIONAL (CCRB) |
| THE RATNAKAR BANK |
| RESERVE BANK OF INDIA |
| RAJKOT NAGARIK SAHAKARI BANK LTD |
| RAJGURUNAGAR SAHAKARI BANK LIMITED |
| THE RAJASTHAN STATE CO-OP BANK |
| SBERBANK |
| SAHEBRAO DESHMUKH COOPERATIVE BANK LIMITED |
| STATE BANK OF BIKANER AND JAIPUR |
| STATE BANK OF HYDERABAD |
| STATE BANK OF INDIA |
| STATE BANK OF MYSORE |
| SAMARTH SAHAKARI BANK LTD |
| STATE BANK OF TRAVANCORE |
| STANDARD CHARTERED BANK |
| THE SURAT DISTRICT CO-OP BAN |
| SHINHAN BANK |
| SHIKSHAK SAHAKARI BANK LIMITED |

| |
|---|
| SOUTH INDIAN BANK |
| SOLAPUR JANATA SAHAKARI BANK LIMITED |
| SUMITOMO MITSUI BANKING CORPORATION |
| SHIVALIK MERCANTILE CO OPERATIVE BANK LTD |
| SOCIETE GENERALE |
| THE SURAT PEOPLE?S CO-OP BANK |
| THE SARASWAT CO-OPERATIVE BANK |
| STATE BANK OF PATIALA |
| STATE BANK OF MAURITIUS |
| SURAT NATIONAL COOPERATIVE BANK LIMITED |
| THE SUTEX COOPERATIVE BANK |
| THE SEVA VIKAS COOPERATIVE BANK LIMITED |
| THE SHAMRAO VITHAL COOP BANK |
| SYNDICATE BANK |
| THANE BHARAT SAHAKARI BANK LTD |
| THE THANE DISTRICT CENTRAL COOPERATIVE BANK LIMITED |
| TUMKUR GRAIN MERCHANTS CO-OP BANK |
| THE THANE JANATA SAHAKARI BANK |
| TAMILNADU MERC. BANK |
| THE TAMILDADU STATE APEX COOP BANK |
| UNION BANK OF INDIA |
| UBS AG |
| UCO BANK |
| UNITED OVERSEAS BANK LIMITED |
| UNITED BANK OF INDIA |
| AXIS BANK |
| THE VARACHHA CO-OP. BANK LTD. |
| VIJAYA BANK |
| THE VISHWESHWAR SAHAKARI BANK LTD |
| VASAI VIKAS SAHAKARI BANK |
| ING VYSYA BANK |
| THE WEST BENGAL STATE CO-OP BANK |
| WESTPAC BANKING CORPORATION |
| YES BANK |
| THE ZOROASTRIAN COOPERATIVE BANK LIMITED |
| ZILA SAHAKRI BANK LIMITED GHAZIABAD |
| Paytm Payments Bank Ltd. |

**For ARS currency:**

| Bank Name |
|---|
| CVU Account |
| Banco de Galicia Y Buenos Aires |
| Banco de La Nacion Argentina |
| Banco de La Provincia de Buenos Aires |
| Industrial and Commercial Bank of China (ICBC) Argentina |
| BBVA |
| Banco de La Provincia de Cordoba |
| Banco Supervielle S.A. |
| Banco de La Ciudad de Buenos Aires |
| Banco Patagonia Sudameris |
| Banco Hipotecario |
| Banco de San Juan |
| Banco Municipal de Rosario |
| Banco Santander |
| Banco Del Chubut |
| Banco de Santa Cruz |
| Banco de La Pampa Sociedad de Economia M |
| Banco de Corrientes |
| Banco Provincia Del Neuquen |
| Brubank S.A.U. |

| |
|---|
| Banco B. I. Creditanstalt |
| HSBC Bank Argentina |
| J P Morgan Chase Bank Sucursal Buenos Aires |
| Banco Credicoop Coop. L |
| Banco de Valores |
| Banco Roela |
| Banco Mariva |
| Banco Itau |
| Bank Of America, National Associa |
| Bnp Paribas |
| Banco Provincia de Tierra Del Fuego |
| Banco de La Republica Oriental Del Uruguay |
| Banco Saenz |
| Banco Meridian |
| Banco Macro |
| Banco Comafi |
| Banco de Inversion Y Comercio Exterior |
| Banco Piano |
| Banco Julio |
| Nuevo Banco de La Rioja |
| Banco Del Sol |
| Nuevo Banco Del Chaco |
| BANCO VOII S.A. |
| Banco de Formosa |
| Banco CMF |
| Banco de Santiago Del Estero |
| Nuevo Banco Industrial de Azul |
| Deutsche Bank |
| Nuevo Banco de Santa Fe |
| Banco Cetelem Argentina |
| Banco de Servicios Financieros |
| Banco Cofidis |
| Banco Bradesco Argentina |
| Banco de Servicios Y Transacciones |
| RCI Banque Argentina |
| Bacs Banco de Credito Y Securitizacion |
| Banco Mas Ventas |
| Wilobank S.A. |
| Nuevo Banco de Entre Rios |
| Banco Columbia |
| Banco Bica S.A. |
| Banco Coinag S.A. |
| Banco de Comercio S.A. |
| Banco Sucredito Regional S.A.U. |
| Banco Dino S.A. |
| Bank of Chine Limited Sucursal Buenos Aires |

**FOR BRL CURRENCY**

| Bank Code | Bank Name |
|---|---|
| 001 | BANCO DO BRASIL S.A. |
| 003 | BANCO DA AMAZONIA S.A. |
| 004 | BANCO DO NORDESTE DO BRASIL S.A. |
| 007 | BANCO NACIONAL DE DESENVOLVIMENTO ECONOMICO E SOCIAL |
| 010 | CREDICOAMO CREDITO RURAL COOPERATIVA |
| 011 | CREDIT SUISSE HEDGING-GRIFFO CORRETORA DE VALORES S.A |
| 012 | BANCO INBURSA S.A. |
| 014 | STATE STREET BRASIL S.A. - BANCO COMERCIAL |
| 015 | UBS BRASIL CORRETORA DE CÂMBIO, TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 016 | COOPERATIVA DE CRÉDITO MÚTUO DOS DESPACHANTES DE TRÂNSITO DE SANTA CATARINA E RI |
| 017 | BNY MELLON BANCO S.A. |

| | |
|---|---|
| 018 | BANCO TRICURY S.A. |
| 021 | BANESTES S.A. BANCO DO ESTADO DO ESPIRITO SANTO |
| 024 | BANCO BANDEPE S.A. |
| 025 | BANCO ALFA S.A. |
| 029 | BANCO ITAÚ CONSIGNADO S.A. |
| 033 | BANCO SANTANDER (BRASIL) S.A. |
| 036 | BANCO BRADESCO BBI S.A. |
| 037 | BANCO DO ESTADO DO PARÁ S.A. |
| 040 | BANCO CARGILL S.A. |
| 041 | BANCO DO ESTADO DO RIO GRANDE DO SUL S.A. |
| 047 | BANCO DO ESTADO DE SERGIPE S.A. |
| 060 | CONFIDENCE CORRETORA DE CÂMBIO S.A. |
| 062 | HIPERCARD BANCO MÚLTIPLO S.A. |
| 063 | BANCO BRADESCARD S.A. |
| 064 | GOLDMAN SACHS DO BRASIL BANCO MULTIPLO S.A. |
| 065 | BANCO ANDBANK (BRASIL) S.A. |
| 066 | BANCO MORGAN STANLEY S.A. |
| 069 | BANCO CREFISA S.A. |
| 070 | BRB - BANCO DE BRASILIA S.A. |
| 074 | BANCO J. SAFRA S.A. |
| 075 | BANCO ABN AMRO S.A. |
| 076 | BANCO KDB DO BRASIL S.A. |
| 077 | BANCO INTER S.A. |
| 078 | HAITONG BANCO DE INVESTIMENTO DO BRASIL S.A. |
| 079 | PICPAY BANK - BANCO MÚLTIPLO S.A |
| 080 | B&T CORRETORA DE CAMBIO LTDA. |
| 081 | BANCOSEGURO S.A. |
| 082 | BANCO TOPÁZIO S.A. |
| 083 | BANCO DA CHINA BRASIL S.A. |
| 084 | UNIPRIME DO BRASIL - COOPERATIVA DE CRÉDITO |
| 085 | COOPERATIVA CENTRAL DE CRÉDITO - AILOS |
| 088 | BANCO RANDON S.A. |
| 089 | CREDISAN COOPERATIVA DE CRÉDITO |
| 093 | PÓLOCRED SOCIEDADE DE CRÉDITO AO MICROEMPREENDEDOR E À EMPRESA DE PEQUENO PORT |
| 094 | BANCO FINAXIS S.A. |
| 095 | TRAVELEX BANCO DE CÂMBIO S.A. |
| 096 | BANCO B3 S.A. |
| 097 | CREDISIS - CENTRAL DE COOPERATIVAS DE CRÉDITO LTDA. |
| 098 | CREDIALIANÇA COOPERATIVA DE CRÉDITO RURAL |
| 099 | UNIPRIME CENTRAL NACIONAL - CENTRAL NACIONAL DE COOPERATIVA DE CREDITO |
| 100 | PLANNER CORRETORA DE VALORES S.A. |
| 101 | RENASCENCA DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA |
| 102 | XP INVESTIMENTOS CORRETORA DE CÂMBIO,TÍTULOS E VALORES MOBILIÁRIOS S/A |
| 104 | CAIXA ECONOMICA FEDERAL |
| 105 | LECCA CRÉDITO, FINANCIAMENTO E INVESTIMENTO S/A |
| 107 | BANCO BOCOM BBM S.A. |
| 111 | OLIVEIRA TRUST DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIARIOS S.A. |
| 113 | NEON CORRETORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 114 | CENTRAL COOPERATIVA DE CRÉDITO NO ESTADO DO ESPÍRITO SANTO - CECOOP |
| 117 | ADVANCED CORRETORA DE CÂMBIO LTDA |
| 119 | BANCO WESTERN UNION DO BRASIL S.A. |
| 120 | BANCO RODOBENS S.A. |
| 121 | BANCO AGIBANK S.A. |
| 122 | BANCO BRADESCO BERJ S.A. |
| 124 | BANCO WOORI BANK DO BRASIL S.A. |
| 125 | BANCO GENIAL S.A. |
| 126 | BR PARTNERS BANCO DE INVESTIMENTO S.A. |
| 127 | CODEPE CORRETORA DE VALORES E CÂMBIO S.A. |
| 128 | MS BANK S.A. BANCO DE CÂMBIO |
| 129 | UBS BRASIL BANCO DE INVESTIMENTO S.A. |

| | |
|---|---|
| 130 | CARUANA S.A. - SOCIEDADE DE CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 131 | TULLETT PREBON BRASIL CORRETORA DE VALORES E CÂMBIO LTDA |
| 132 | ICBC DO BRASIL BANCO MÚLTIPLO S.A. |
| 133 | CONFEDERAÇÃO NACIONAL DAS COOPERATIVAS CENTRAIS DE CRÉDITO E ECONOMIA FAMILIAR E |
| 134 | BGC LIQUIDEZ DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA |
| 136 | CONFEDERAÇÃO NACIONAL DAS COOPERATIVAS CENTRAIS UNICRED LTDA. - UNICRED DO BRASI |
| 138 | GET MONEY CORRETORA DE CÂMBIO S.A. |
| 139 | INTESA SANPAOLO BRASIL S.A. - BANCO MÚLTIPLO |
| 140 | NU INVEST CORRETORA DE VALORES S.A. |
| 142 | BROKER BRASIL CORRETORA DE CÂMBIO LTDA. |
| 143 | TREVISO CORRETORA DE CÂMBIO S.A. |
| 144 | BEXS BANCO DE CÂMBIO S/A |
| 145 | LEVYCAM - CORRETORA DE CAMBIO E VALORES LTDA. |
| 146 | GUITTA CORRETORA DE CAMBIO LTDA. |
| 149 | FACTA FINANCEIRA S.A. - CRÉDITO FINANCIAMENTO E INVESTIMENTO |
| 157 | ICAP DO BRASIL CORRETORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 159 | CASA DO CRÉDITO S.A. SOCIEDADE DE CRÉDITO AO MICROEMPREENDEDOR |
| 163 | COMMERZBANK BRASIL S.A. - BANCO MÚLTIPLO |
| 173 | BRL TRUST DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 174 | PEFISA S.A. - CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 177 | GUIDE INVESTIMENTOS S.A. CORRETORA DE VALORES |
| 180 | CM CAPITAL MARKETS CORRETORA DE CÂMBIO, TÍTULOS E VALORES MOBILIÁRIOS LTDA |
| 183 | SOCRED S.A. - SOCIEDADE DE CRÉDITO AO MICROEMPREENDEDOR E À EMPRESA DE PEQUENO P |
| 184 | BANCO ITAÚ BBA S.A. |
| 188 | ATIVA INVESTIMENTOS S.A. CORRETORA DE TÍTULOS, CÂMBIO E VALORES |
| 189 | HS FINANCEIRA S/A CREDITO, FINANCIAMENTO E INVESTIMENTOS |
| 190 | SERVICOOP - COOPERATIVA DE CRÉDITO DOS SERVIDORES PÚBLICOS ESTADUAIS E MUNICIPAI |
| 191 | NOVA FUTURA CORRETORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 194 | PARMETAL DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA |
| 195 | VALOR SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 196 | FAIR CORRETORA DE CAMBIO S.A. |
| 197 | STONE INSTITUIÇÃO DE PAGAMENTO S.A. |
| 208 | BANCO BTG PACTUAL S.A. |
| 212 | BANCO ORIGINAL S.A. |
| 213 | BANCO ARBI S.A. |
| 217 | BANCO JOHN DEERE S.A. |
| 218 | BANCO BS2 S.A. |
| 222 | BANCO CRÉDIT AGRICOLE BRASIL S.A. |
| 224 | BANCO FIBRA S.A. |
| 233 | BANCO CIFRA S.A. |
| 237 | BANCO BRADESCO S.A. |
| 241 | BANCO CLASSICO S.A. |
| 243 | BANCO MASTER S/A |
| 246 | BANCO ABC BRASIL S.A. |
| 249 | BANCO INVESTCRED UNIBANCO S.A. |
| 250 | BCV - BANCO DE CRÉDITO E VAREJO S.A. |
| 253 | BEXS CORRETORA DE CÂMBIO S/A |
| 254 | PARANÁ BANCO S.A. |
| 259 | MONEYCORP BANCO DE CÂMBIO S.A. |
| 260 | NU PAGAMENTOS S.A. - INSTITUIÇÃO DE PAGAMENTO |
| 265 | BANCO FATOR S.A. |
| 266 | BANCO CEDULA S.A. |
| 268 | BARI COMPANHIA HIPOTECÁRIA |
| 269 | BANCO HSBC S.A. |
| 270 | SAGITUR CORRETORA DE CÂMBIO S.A. |
| 271 | IB CORRETORA DE CÂMBIO, TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 272 | AGK CORRETORA DE CAMBIO S.A. |
| 273 | COOPERATIVA DE CRÉDITO RURAL DE SÃO MIGUEL DO OESTE - SULCREDI/SÃO MIGUEL |
| 274 | BMP SOCIEDADE DE CRÉDITO AO MICROEMPREENDEDOR E A EMPRESA DE PEQUENO PORTE LTDA. |

| | |
|---|---|
| 276 | BANCO SENFF S.A. |
| 278 | GENIAL INVESTIMENTOS CORRETORA DE VALORES MOBILIÁRIOS S.A. |
| 279 | PRIMACREDI COOPERATIVA DE CRÉDITO DE PRIMAVERA DO LESTE |
| 280 | WILL FINANCEIRA S.A. CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 281 | COOPERATIVA DE CRÉDITO RURAL COOPAVEL |
| 283 | RB INVESTIMENTOS DISTRIBUIDORA DE TITULOS E VALORES MOBILIARIOS LIMITADA |
| 285 | FRENTE CORRETORA DE CÂMBIO LTDA. |
| 286 | UNIPRIME OURO - COOPERATIVA DE CRÉDITO DE OURO |
| 288 | CAROL DISTRIBUIDORA DE TITULOS E VALORES MOBILIARIOS LTDA. |
| 289 | EFX CORRETORA DE CÂMBIO LTDA. |
| 290 | PAGSEGURO INTERNET INSTITUIÇÃO DE PAGAMENTO S.A. |
| 292 | BS2 DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 293 | LASTRO RDV DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 296 | OZ CORRETORA DE CÂMBIO S.A. |
| 298 | VIP'S CORRETORA DE CÂMBIO LTDA. |
| 299 | BANCO AFINZ S.A. - BANCO MÚLTIPLO |
| 300 | BANCO DE LA NACION ARGENTINA |
| 301 | DOCK INSTITUIÇÃO DE PAGAMENTO S.A. |
| 306 | PORTOPAR DISTRIBUIDORA DE TITULOS E VALORES MOBILIARIOS LTDA. |
| 307 | TERRA INVESTIMENTOS DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 309 | CAMBIONET CORRETORA DE CÂMBIO LTDA. |
| 310 | VORTX DISTRIBUIDORA DE TITULOS E VALORES MOBILIARIOS LTDA. |
| 311 | DOURADA CORRETORA DE CÂMBIO LTDA. |
| 312 | HSCM - SOCIEDADE DE CRÉDITO AO MICROEMPREENDEDOR E À EMPRESA DE PEQUENO PORTE LT |
| 313 | AMAZÔNIA CORRETORA DE CÂMBIO LTDA. |
| 318 | BANCO BMG S.A. |
| 319 | OM DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA |
| 320 | CHINA CONSTRUCTION BANK (BRASIL) BANCO MÚLTIPLO S/A |
| 321 | CREFAZ SOCIEDADE DE CRÉDITO AO MICROEMPREENDEDOR E A EMPRESA DE PEQUENO PORTE LT |
| 322 | COOPERATIVA DE CRÉDITO RURAL DE ABELARDO LUZ - SULCREDI/CREDILUZ |
| 323 | MERCADO PAGO INSTITUIÇÃO DE PAGAMENTO LTDA. |
| 324 | CARTOS SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 325 | ÓRAMA DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 326 | PARATI - CREDITO, FINANCIAMENTO E INVESTIMENTO S.A. |
| 328 | COOPERATIVA DE ECONOMIA E CRÉDITO MÚTUO DOS FABRICANTES DE CALÇADOS DE SAPIRANGA |
| 329 | QI SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 330 | BANCO BARI DE INVESTIMENTOS E FINANCIAMENTOS S.A. |
| 331 | FRAM CAPITAL DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 332 | ACESSO SOLUÇÕES DE PAGAMENTO S.A. - INSTITUIÇÃO DE PAGAMENTO |
| 334 | BANCO BESA S.A. |
| 335 | BANCO DIGIO S.A. |
| 336 | BANCO C6 S.A. |
| 340 | SUPERDIGITAL INSTITUIÇÃO DE PAGAMENTO S.A. |
| 341 | ITAÚ UNIBANCO S.A. |
| 342 | CREDITAS SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 343 | FFA SOCIEDADE DE CRÉDITO AO MICROEMPREENDEDOR E À EMPRESA DE PEQUENO PORTE LTDA. |
| 348 | BANCO XP S.A. |
| 349 | AL5 S.A. CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 350 | COOPERATIVA DE CRÉDITO RURAL DE PEQUENOS AGRICULTORES E DA REFORMA AGRÁRIA DO CE |
| 352 | TORO CORRETORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 355 | ÓTIMO SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 358 | MIDWAY S.A. - CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 359 | ZEMA CRÉDITO, FINANCIAMENTO E INVESTIMENTO S/A |
| 360 | TRINUS CAPITAL DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 362 | CIELO S.A. - INSTITUIÇÃO DE PAGAMENTO |
| 363 | SINGULARE CORRETORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 364 | EFÍ S.A. - INSTITUIÇÃO DE PAGAMENTO |
| 365 | SIMPAUL CORRETORA DE CAMBIO E VALORES MOBILIARIOS S.A. |
| 366 | BANCO SOCIETE GENERALE BRASIL S.A. |
| 367 | VITREO DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |

| | |
|---|---|
| 368 | BANCO CSF S.A. |
| 370 | BANCO MIZUHO DO BRASIL S.A. |
| 371 | WARREN CORRETORA DE VALORES MOBILIÁRIOS E CÂMBIO LTDA. |
| 373 | UP.P SOCIEDADE DE EMPRÉSTIMO ENTRE PESSOAS S.A. |
| 374 | REALIZE CRÉDITO, FINANCIAMENTO E INVESTIMENTO S.A. |
| 376 | BANCO J.P. MORGAN S.A. |
| 377 | BMS SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 378 | BANCO BRASILEIRO DE CRÉDITO SOCIEDADE ANÔNIMA |
| 379 | COOPERFORTE - COOPERATIVA DE ECONOMIA E CRÉDITO MÚTUO DE FUNCIONÁRIOS DE INSTITU |
| 380 | PICPAY INSTITUIÇÃO DE PAGAMENTO S.A. |
| 381 | BANCO MERCEDES-BENZ DO BRASIL S.A. |
| 382 | FIDÚCIA SOCIEDADE DE CRÉDITO AO MICROEMPREENDEDOR E À EMPRESA DE PEQUENO PORTE L |
| 383 | EBANX INSTITUICAO DE PAGAMENTOS LTDA. |
| 384 | GLOBAL FINANÇAS SOCIEDADE DE CRÉDITO AO MICROEMPREENDEDOR E À EMPRESA DE PEQUENO |
| 385 | COOPERATIVA DE ECONOMIA E CREDITO MUTUO DOS TRABALHADORES PORTUARIOS DA GRANDE V |
| 386 | NU FINANCEIRA S.A. - SOCIEDADE DE CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 387 | BANCO TOYOTA DO BRASIL S.A. |
| 389 | BANCO MERCANTIL DO BRASIL S.A. |
| 390 | BANCO GM S.A. |
| 391 | COOPERATIVA DE CREDITO RURAL DE IBIAM - SULCREDI/IBIAM |
| 393 | BANCO VOLKSWAGEN S.A. |
| 394 | BANCO BRADESCO FINANCIAMENTOS S.A. |
| 395 | F.D'GOLD - DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 396 | HUB INSTITUIÇÃO DE PAGAMENTO S.A. |
| 397 | LISTO SOCIEDADE DE CREDITO DIRETO S.A. |
| 398 | IDEAL CORRETORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 399 | KIRTON BANK S.A. - BANCO MÚLTIPLO |
| 400 | COOPERATIVA DE CRÉDITO, POUPANÇA E SERVIÇOS FINANCEIROS DO CENTRO OESTE - CREDIT |
| 401 | IUGU INSTITUIÇÃO DE PAGAMENTO S.A. |
| 402 | COBUCCIO S/A - SOCIEDADE DE CRÉDITO, FINANCIAMENTO E INVESTIMENTOS |
| 403 | CORA SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 404 | SUMUP SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 406 | ACCREDITO - SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 407 | ÍNDIGO INVESTIMENTOS DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 408 | BONUSPAGO SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 410 | PLANNER SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 411 | VIA CERTA FINANCIADORA S.A. - CRÉDITO, FINANCIAMENTO E INVESTIMENTOS |
| 412 | SOCIAL BANK BANCO MÚLTIPLO S/A |
| 413 | BANCO BV S.A. |
| 414 | LEND SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 416 | LAMARA SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 418 | ZIPDIN SOLUÇÕES DIGITAIS SOCIEDADE DE CRÉDITO DIRETO S/A |
| 419 | NUMBRS SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 421 | LAR COOPERATIVA DE CRÉDITO - LAR CREDI |
| 422 | BANCO SAFRA S.A. |
| 423 | COLUNA S/A DISTRIBUIDORA DE TITULOS E VALORES MOBILIÁRIOS |
| 425 | SOCINAL S.A. - CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 426 | BIORC FINANCEIRA - CRÉDITO, FINANCIAMENTO E INVESTIMENTO S.A. |
| 427 | COOPERATIVA DE CREDITO DOS SERVIDORES DA UNIVERSIDADE FEDERAL DO ESPIRITO SANTO |
| 428 | CREDSYSTEM SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 429 | CREDIARE S.A. - CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 430 | COOPERATIVA DE CREDITO RURAL SEARA - CREDISEARA |
| 433 | BR-CAPITAL DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 435 | DELCRED SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 438 | TRUSTEE DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 439 | ID CORRETORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 440 | CREDIBRF - COOPERATIVA DE CRÉDITO |
| 442 | MAGNETIS - DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA |
| 443 | CREDIHOME SOCIEDADE DE CRÉDITO DIRETO S.A. |

| | |
|---|---|
| 444 | TRINUS SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 445 | PLANTAE S.A. - CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 447 | MIRAE ASSET WEALTH MANAGEMENT (BRAZIL) CORRETORA DE CÂMBIO, TÍTULOS E VALORES MO |
| 448 | HEMERA DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 449 | DM SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 450 | FITBANK INSTITUIÇÃO DE PAGAMENTOS ELETRÔNICOS S.A. |
| 451 | J17 - SOCIEDADE DE CRÉDITO DIRETO S/A |
| 452 | CREDIFIT SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 454 | MÉRITO DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 455 | FÊNIX DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 456 | BANCO MUFG BRASIL S.A. |
| 457 | UY3 SOCIEDADE DE CRÉDITO DIRETO S/A |
| 458 | HEDGE INVESTMENTS DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 459 | COOPERATIVA DE CRÉDITO MÚTUO DE SERVIDORES PÚBLICOS DO ESTADO DE SÃO PAULO - CRE |
| 460 | UNAVANTI SOCIEDADE DE CRÉDITO DIRETO S/A |
| 461 | ASAAS GESTÃO FINANCEIRA INSTITUIÇÃO DE PAGAMENTO S.A. |
| 462 | STARK SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 463 | AZUMI DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 464 | BANCO SUMITOMO MITSUI BRASILEIRO S.A. |
| 465 | CAPITAL CONSIG SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 467 | MASTER S/A CORRETORA DE CÂMBIO, TÍTULOS E VALORES MOBILIÁRIOS |
| 468 | PORTOSEG S.A. - CREDITO, FINANCIAMENTO E INVESTIMENTO |
| 469 | LIGA INVEST DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA |
| 470 | CDC SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 471 | COOPERATIVA DE ECONOMIA E CREDITO MUTUO DOS SERVIDORES PUBLICOS DE PINHÃO - CRES |
| 473 | BANCO CAIXA GERAL - BRASIL S.A. |
| 475 | BANCO YAMAHA MOTOR DO BRASIL S.A. |
| 477 | CITIBANK N.A. |
| 478 | GAZINCRED S.A. SOCIEDADE DE CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 479 | BANCO ITAUBANK S.A. |
| 481 | SUPERLÓGICA SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 482 | SBCASH SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 484 | MAF DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A. |
| 487 | DEUTSCHE BANK S.A. - BANCO ALEMAO |
| 488 | JPMORGAN CHASE BANK, NATIONAL ASSOCIATION |
| 495 | BANCO DE LA PROVINCIA DE BUENOS AIRES |
| 505 | BANCO CREDIT SUISSE (BRASIL) S.A. |
| 506 | RJI CORRETORA DE TITULOS E VALORES MOBILIARIOS LTDA |
| 507 | SOCIEDADE DE CRÉDITO, FINANCIAMENTO E INVESTIMENTO EFÍ S.A. |
| 508 | AVENUE SECURITIES DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 509 | CELCOIN INSTITUICAO DE PAGAMENTO S.A. |
| 510 | FFCRED SOCIEDADE DE CRÉDITO DIRETO S.A.. |
| 511 | MAGNUM SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 512 | FINVEST DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 513 | ATF CREDIT SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 516 | FC FINANCEIRA S.A. - CRÉDITO, FINANCIAMENTO E INVESTIMENTO |
| 518 | MERCADO CRÉDITO SOCIEDADE DE CRÉDITO, FINANCIAMENTO E INVESTIMENTO S.A. |
| 519 | LIONS TRUST DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS LTDA. |
| 521 | PEAK SOCIEDADE DE EMPRÉSTIMO ENTRE PESSOAS S.A. |
| 523 | HR DIGITAL - SOCIEDADE DE CRÉDITO DIRETO S/A |
| 525 | INTERCAM CORRETORA DE CÂMBIO LTDA. |
| 526 | MONETARIE SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 527 | ATICCA - SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 529 | PINBANK BRASIL INSTITUIÇÃO DE PAGAMENTO S.A. |
| 535 | MARÚ SOCIEDADE DE CRÉDITO DIRETO S.A. |
| 536 | NEON PAGAMENTOS S.A. - INSTITUIÇÃO DE PAGAMENTO |
| 545 | SENSO CORRETORA DE CAMBIO E VALORES MOBILIARIOS S.A |
| 600 | BANCO LUSO BRASILEIRO S.A. |
| 604 | BANCO INDUSTRIAL DO BRASIL S.A. |

| | |
|---|---|
| 610 | BANCO VR S.A. |
| 611 | BANCO PAULISTA S.A. |
| 612 | BANCO GUANABARA S.A. |
| 613 | OMNI BANCO S.A. |
| 623 | BANCO PAN S.A. |
| 626 | BANCO C6 CONSIGNADO S.A. |
| 630 | BANCO LETSBANK S.A. |
| 633 | BANCO RENDIMENTO S.A. |
| 634 | BANCO TRIANGULO S.A. |
| 637 | BANCO SOFISA S.A. |
| 643 | BANCO PINE S.A. |
| 653 | BANCO VOITER S.A. |
| 654 | BANCO DIGIMAIS S.A. |
| 655 | BANCO VOTORANTIM S.A. |
| 707 | BANCO DAYCOVAL S.A. |
| 712 | BANCO OURINVEST S.A. |
| 720 | BANCO RNX S.A. |
| 739 | BANCO CETELEM S.A. |
| 741 | BANCO RIBEIRAO PRETO S.A. |
| 743 | BANCO SEMEAR S.A. |
| 745 | BANCO CITIBANK S.A. |
| 746 | BANCO MODAL S.A. |
| 747 | BANCO RABOBANK INTERNATIONAL BRASIL S.A. |
| 748 | BANCO COOPERATIVO SICREDI S.A. |
| 751 | SCOTIABANK BRASIL S.A. BANCO MÚLTIPLO |
| 752 | BANCO BNP PARIBAS BRASIL S.A. |
| 753 | NOVO BANCO CONTINENTAL S.A. - BANCO MÚLTIPLO |
| 754 | BANCO SISTEMA S.A. |
| 755 | BANK OF AMERICA MERRILL LYNCH BANCO MÚLTIPLO S.A. |
| 756 | BANCO COOPERATIVO SICOOB S.A. - BANCO SICOOB |
| 757 | BANCO KEB HANA DO BRASIL S.A. |

**For CLP currency:**

| Bank Name |
|---|
| Banco de Chile |
| Banco Internacional |
| Banco del Estado de Chile |
| Scotiabank Chile |
| Banco Crédito e Inversiones |
| Banco Bice |
| HSBC Bank |
| Banco Santander- Santiago |
| Itau Corpbanca |
| Banco Security |
| Banco Falabella |
| Banco Ripley |
| Banco Consorcio |
| BBVA Chile |
| Banco del Desarrollo |
| Coopeuch |
| Prepago los Héroes |
| Tenpo Prepago |

**For COP currency:**

| Bank Name |
|---|
| BANCO DE BOGOTA |
| BANCO POPULAR |
| BANCO SANTANDER |
| BANCOLOMBIA |
| HSBC |
| BANCO SUDAMERIS |

| |
|---|
| BBVA |
| ITAU |
| BANCO COLPATRIA |
| BANCO DE OCCIDENTE |
| BANCOLDEX S.A. |
| BANCO CAJA SOCIAL BCSC |
| BANCO AGRARIO |
| BANCO MUNDO MUJER |
| BANCO DAVIVIENDA |
| BANCO AV VILLAS |
| BANCO W S.A |
| BANCO PROCREDIT |
| BANCAMIA S.A |
| BANCO PICHINCHA |
| BANCOOMEVA |
| BANCO FALABELLA S.A |
| BANCO FINANDINA S.A. |
| BANCO MULTIBANK S.A. |
| BANCO SERFINANZA S.A. |
| COOPCENTRAL S.A |
| COOPERATIVA FINANCIERA DE ANTIOQUIA |
| COTRAFA COOPERATIVA FINANCIERA |
| CONFIAR |
| FINANCIERA JURISCOOP |
| COLTEFINANCIERA S.A. |
| NEQUI |

**For MXN currency:**

| Bank Name |
|---|
| BANAMEX |
| BANCOMEXT |
| BANOBRAS |
| BBVA BANCOMER |
| SANTANDER |
| BANJERCITO |
| HSBC |
| BAJIO |
| IXE |
| INBURSA |
| INTERACCIONES |
| MIFEL |
| SCOTIABANK |
| BANREGIO |
| INVEX |
| BANSI |
| AFIRME |
| BANORTE |
| THE ROYAL BANK |
| AMERICAN EXPRESS |
| BAMSA |
| TOKYO |
| JP MORGAN |
| BMONEX |
| VE POR MAS |
| ING |
| DEUTSCHE |
| CREDIT SUISSE |
| AZTECA |
| AUTOFIN |
| BARCLAYS |

| |
|---|
| COMPARTAMOS |
| BANCO FAMSA |
| BMULTIVA |
| ACTINVER |
| WALMART |
| NAFIN |
| INTERBANCO |
| BANCOPPEL |
| ABC CAPITAL |
| UBS BANK |
| CONSUBANCO |
| VOLKSWAGEN |
| CIBANCO |
| BBASE |
| BANSEFI |
| HIPOTECARIA FEDERAL |
| MONEXCB |
| GBM |
| MASARI |
| VALUE |
| ESTRUCTURADORES |
| TIBER |
| VECTOR |
| B&B |
| ACCIVAL |
| MERRILL LYNCH |
| FINAMEX |
| VALMEX |
| UNICA |
| MAPFRE |
| PROFUTURO |
| CB ACTINVER |
| OACTIN |
| SKANDIA |
| CBDEUTSCHE |
| ZURICH |
| ZURICHVI |
| SU CASITA |
| CB INTERCAM |
| CI BOLSA |
| BULLTICK CB |
| STERLING |
| FINCOMUN |
| HDI SEGUROS |
| ORDER |
| AKALA |
| CB JPMORGAN |
| REFORMA |
| STP |
| TELECOMM |
| EVERCORE |
| SKANDIA |
| SEGMTY |
| ASEA |
| KUSPIT |
| SOFIEXPRESS |
| UNAGRA |
| OPCIONES EMPRESARIALES DEL NOROESTE |
| LIBERTAD |
| CLS |

| |
|---|
| INDEVAL |

**For PEN currency:**

| Bank Name |
|---|
| Banco Central de Reserva |
| Banco de Crédito del Perú |
| Interbank |
| Citibank |
| Scotiabank |
| BBVA Continental |
| Banco de la Nación |
| Banco de Comercio |
| Banco Financiero |
| Banco Interamericano de Finanzas (BIF) |
| Crediscotia Financiera |
| Mi Banco |
| Banco GNB Perú S.A. |
| Banco Falabella |
| Santander |
| Caja Metropolitana de Lima |
| Caja Municipal de Ahorro y Crédito Piura SAC |
| Caja Municipal de Ahorro y Crédito Trujillo |
| Caja Municipal de Ahorro y Crédito Arequipa |
| Caja Municipal de Ahorro y Crédito Sullana |
| Caja Municipal de Ahorro y Crédito Cuzco |
| Caja Municipal de Ahorro y Crédito Huancayo |
| Caja Municipal de Ahorro y Crédito Tacna |

**For UYU currency:**

| Bank Name |
|---|
| BROU - Banco de la República Oriental del Uruguay |
| Banco Hipotecario del Uruguay |
| Banco Bandes |
| Banco ITAU |
| Scotiabank |
| Banco Santander |
| Banco Bilbao Vizcaya Argentaria |
| HSBC Bank |
| Banque Heritage |
| Citibank N.A. Sucursal |
| Banco de la Nación Argentina |

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bank_payout</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>50000</amount>
  <currency>INR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |

| | | |
|---|---|---|
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bank_payout</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>110</code>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>50000</amount>
  <currency>INR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### BITPAY PAYOUT

BitPay Payout is a crypto currency payout method where merchants are requesting payouts in FIAT currency and the funds are transferred in Bitcoin equivalent to a crypto wallet address.

BitPay Payout is an asynchronous transaction type supported through the Processing API, Virtual Terminal and Web Payment Form.

The payout requests are processed once a day at 11:00 GMT and the settlement usually takes 24 hours.

For amounts greater than 3000 USD or equivalent in other currency, additional KYC authentication might be required by BitPay

**Supported countries**

| Country name | Country code |
|---|---|
| Afghanistan | AF |
| Aland Islands | AX |
| Albania | AL |
| American Samoa | AS |
| Andorra | AD |
| Angola | AO |
| Anguilla | AI |
| Antarctica | AQ |
| Antigua and Barbuda | AG |
| Argentina | AR |
| Armenia | AM |
| Aruba | AW |
| Australia | AU |
| Austria | AT |
| Azerbaijan | AZ |
| Bahamas | BS |
| Bahrain | BH |
| Barbados | BB |
| Belarus | BY |
| Belgium | BE |
| Belize | BZ |

| | |
|---|---|
| Benin | BJ |
| Bermuda | BM |
| Bhutan | BT |
| Bonaire, Sint Eustatius and Saba | BQ |
| Bosnia and Herzegovina | BA |
| Botswana | BW |
| Bouvet Island | BV |
| Brazil | BR |
| British Indian Ocean Territory | IO |
| Brunei Darussalam | BN |
| Bulgaria | BG |
| Burkina Faso | BF |
| Burundi | BI |
| Cameroon | CM |
| Canada | CA |
| Cape Verde | CV |
| Cayman Islands | KY |
| Central African Republic | CF |
| Chad | TD |
| Chile | CL |
| China | CN |
| Christmas Island | CX |
| Cocos (Keeling) Islands | CC |
| Colombia | CO |
| Comoros | KM |
| Congo | CG |
| Congo, the Democratic Republic of the | CD |
| Cook Islands | CK |
| Costa Rica | CR |
| Cote D'Ivoire | CI |
| Croatia | HR |
| Cuba | CU |
| Curacao | CW |
| Cyprus | CY |
| Czech Republic | CZ |
| Denmark | DK |
| Djibouti | DJ |
| Dominica | DM |
| Dominican Republic | DO |
| El Salvador | SV |
| Equatorial Guinea | GQ |
| Eritrea | ER |
| Estonia | EE |
| Ethiopia | ET |
| Falkland Islands (Malvinas) | FK |
| Faroe Islands | FO |
| Fiji | FJ |
| Finland | FI |
| France | FR |
| French Guiana | GF |
| French Polynesia | PF |
| French Southern Territories | TF |
| Gabon | GA |
| Gambia | GM |
| Georgia | GE |
| Germany | DE |
| Ghana | GH |
| Gibraltar | GI |
| Greece | GR |
| Greenland | GL |

| | |
|---|---|
| Grenada | GD |
| Guadeloupe | GP |
| Guam | GU |
| Guatemala | GT |
| Guernsey | GG |
| Guinea | GN |
| Guinea-Bissau | GW |
| Guyana | GY |
| Haiti | HT |
| Heard Island and McDonald Islands | HM |
| Holy See (Vatican City State) | VA |
| Honduras | HN |
| Hong Kong | HK |
| Hungary | HU |
| Iceland | IS |
| India | IN |
| Iran, Islamic Republic of | IR |
| Ireland | IE |
| Isle of Man | IM |
| Israel | IL |
| Italy | IT |
| Jamaica | JM |
| Japan | JP |
| Jersey | JE |
| Jordan | JO |
| Kazakhstan | KZ |
| Kenya | KE |
| Kiribati | KI |
| Korea, Democratic People's Republic of | KP |
| Korea, Republic of | KR |
| Kosovo, Republic of | XK |
| Kuwait | KW |
| Lao People's Democratic Republic | LA |
| Latvia | LV |
| Lebanon | LB |
| Lesotho | LS |
| Liberia | LR |
| Libyan Arab Jamahiriya | LY |
| Liechtenstein | LI |
| Lithuania | LT |
| Luxembourg | LU |
| Macao | MO |
| Madagascar | MG |
| Malawi | MW |
| Malaysia | MY |
| Maldives | MV |
| Mali | ML |
| Malta | MT |
| Marshall Islands | MH |
| Martinique | MQ |
| Mauritania | MR |
| Mauritius | MU |
| Mayotte | YT |
| Mexico | MX |
| Micronesia, Federated States of | FM |
| Moldova, Republic of | MD |
| Monaco | MC |
| Mongolia | MN |
| Montenegro | ME |

| | |
|---|---|
| Montserrat | MS |
| Mozambique | MZ |
| Myanmar | MM |
| Namibia | NA |
| Nauru | NR |
| Netherlands | NL |
| Netherlands Antilles | AN |
| New Caledonia | NC |
| New Zealand | NZ |
| Nicaragua | NI |
| Niger | NE |
| Nigeria | NG |
| Niue | NU |
| Norfolk Island | NF |
| Northern Mariana Islands | MP |
| Norway | NO |
| Oman | OM |
| Palau | PW |
| Panama | PA |
| Papua New Guinea | PG |
| Paraguay | PY |
| Peru | PE |
| Philippines | PH |
| Pitcairn | PN |
| Poland | PL |
| Portugal | PT |
| Puerto Rico | PR |
| Qatar | QA |
| Reunion | RE |
| Romania | RO |
| Russian Federation | RU |
| Rwanda | RW |
| Saint Barthélemy | BL |
| Saint Helena | SH |
| Saint Kitts and Nevis | KN |
| Saint Lucia | LC |
| Saint Martin French Part | MF |
| Saint Pierre and Miquelon | PM |
| Saint Vincent and the Grenadines | VC |
| Samoa | WS |
| San Marino | SM |
| Sao Tome and Principe | ST |
| Saudi Arabia | SA |
| Senegal | SN |
| Serbia | RS |
| Seychelles | SC |
| Sierra Leone | SL |
| Singapore | SG |
| Sint Maarten (Dutch part) | SX |
| Slovakia | SK |
| Slovenia | SI |
| Solomon Islands | SB |
| Somalia | SO |
| South Africa | ZA |
| South Georgia and the South Sandwich Islands | GS |
| South Sudan | SS |
| Spain | ES |
| Sri Lanka | LK |
| Sudan | SD |
| Suriname | SR |

| | |
|---|---|
| Svalbard and Jan Mayen | SJ |
| Swaziland | SZ |
| Sweden | SE |
| Switzerland | CH |
| Syrian Arab Republic | SY |
| Taiwan, Province of China | TW |
| Tajikistan | TJ |
| Tanzania, United Republic of | TZ |
| Thailand | TH |
| Timor-Leste | TL |
| Togo | TG |
| Tokelau | TK |
| Tonga | TO |
| Trinidad and Tobago | TT |
| Tunisia | TN |
| Turkmenistan | TM |
| Turks and Caicos Islands | TC |
| Tuvalu | TV |
| Uganda | UG |
| Ukraine | UA |
| United Arab Emirates | AE |
| United Kingdom | GB |
| United States | US |
| United States Minor Outlying Islands | UM |
| Uruguay | UY |
| Uzbekistan | UZ |
| Vanuatu | VU |
| Venezuela, Bolivarian Republic of | VE |
| Virgin Islands, British | VG |
| Virgin Islands, U.S. | VI |
| Wallis and Futuna | WF |
| Western Sahara | EH |
| Yemen | YE |
| Zambia | ZM |
| Zimbabwe | ZW |

**Supported currencies**

| Currency name | Currency code |
|---|---|
| American Dollar | USD |
| Euro | EUR |

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
  <?xml version="1.0" encoding="UTF-8"?>
  <payment_transaction>
    <transaction_type>bitpay_payout</transaction_type>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <usage>40208 concert tickets</usage>
    <remote_ip>245.253.2.12</remote_ip>
    <notification_url>https://www.example.com/notification</notification_url>
    <return_success_url>http://www.example.com/success</return_success_url>
    <return_failure_url>http://www.example.com/failure</return_failure_url>
    <amount>2000</amount>
    <currency>USD</currency>
    <crypto_address>n1jE32b88mtT7UETWvV6WgrV5AUVTPDxW8</crypto_address>
    <crypto_wallet_provider>other</crypto_wallet_provider>
    <customer_email>travis@example.com</customer_email>
  </payment_transaction>'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
  <?xml version="1.0" encoding="UTF-8"?>
  <payment_transaction>
    <transaction_type>bitpay_payout</transaction_type>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <usage>40208 concert tickets</usage>
    <remote_ip>245.253.2.12</remote_ip>
    <notification_url>https://www.example.com/notification</notification_url>
    <return_success_url>http://www.example.com/success</return_success_url>
    <return_failure_url>http://www.example.com/failure</return_failure_url>
    <amount>500000</amount>
    <currency>USD</currency>
    <crypto_address>n1jE32b88mtT7UETWvV6WgrV5AUVTPDxW7</crypto_address>
```

```
    <crypto_wallet_provider>kraken</crypto_wallet_provider>
    <billing_address>
      <first_name>Travis</first_name>
      <last_name>Pastrana</last_name>
      <address1>Muster Str. 12</address1>
      <zip_code>10178</zip_code>
      <city>Los Angeles</city>
      <neighborhood>Hollywood</neighborhood>
      <state>CA</state>
      <country>US</country>
    </billing_address>
  </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **bitpay_payout** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| notification_url | required | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| crypto_address | required | string(255) | Valid crypto address where the funds will be received |
| crypto_wallet_provider | required | string(255) | If crypto wallet provider is not in the table below, you must send 'other' |
| **billing_address** | required* | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Crypto wallet providers with additional requirements**

| crypto wallet provider | description | website |
|---|---|---|
| BitGo | crypto wallet provider | official website |
| Uphold | crypto wallet provider | official website |
| Circle | crypto exchange | official website |
| Coinbase | crypto wallet provider | official website |
| GDax | crypto exchange | official website |
| Gemini | crypto exchange | official website |
| ITBit | crypto exchange | official website |
| Kraken | crypto exchange | official website |

Address fields are required in case the crypto wallet provider is in the list above and the payout amount is greater than 3000 USD or the equivalent in other currency

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bitpay_payout</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
```

```
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>500000</amount>
  <currency>USD</currency>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bitpay_payout</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <payment_response>response_code</payment_response>
  <code>940</code>
  <technical_message>Bitcoin address is invalid</technical_message>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>2000</amount>
  <currency>USD</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

#### EZEECARD PAYOUT

eZeeCard Payout is a sync based payout method. It's merchant initiated and can only reference specific transaction types:

- Capture
- Sale
- Sale3d
- InitRecurringSale
- InitRecurringSale3d
- RecurringSale

Those need to have been completed using a card issued by our Issuing API.

> eZeeCard Payout is available through Processing API and VT only!

> eZeeCard Payout has amount limits of minimum 10 EUR and maximum of 800 EUR per transaction or its equivalent in other currencies

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>ezeecard_payout</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>1000</amount>
   <currency>EUR</currency>
   <reference_id>43672</reference_id>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **ezeecard_payout** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ezeecard_payout</transaction_type>
  <status>approved</status>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <amount>1000</amount>
  <currency>EUR</currency>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>ezeecard_payout</transaction_type>
  <status>error</status>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <technical_message>invalid deposit transaction</technical_message>
  <payment_response>message</payment_response>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:44Z</timestamp>
  <amount>1000</amount>
  <currency>EUR</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

#### PAYOUT

Payouts are credits without a reference transaction and as such are highly regulated and need specific gateway terminal configuration, so be sure to contact the IT Support team at tech-support@e-comprocessing.com in case you want Payouts to be enabled. A valid bank account number needs to be provided.

Using a payout, the amount is billed to the customer’s credit card. It can be reversed via a void transaction on the same day of the transaction.

Both Visa and Mastercard/Maestro Payouts are authorized real-time.

Note that for exceptional cases with some countries Visa OCTS will not be authorized through the schemes but batched for offline settlement on the same day. This means that the authorization code and issuer response code will not be available only for them.

Note that VISA OCT transactions with Australian or Canadian card bins will require the merchant zip code to be set, either through the dynamic descriptor parameter or through the merchant configuration.

This transaction type supports Tokenization.

This transaction type supports Account Name Inquiry attributes.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>payout</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
   </account_owner>
   <credential_on_file_transaction_identifier>123456789101112</credential_on_file_transaction_identifier>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **payout** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| crypto | optional | "true" | Signifies whether a crypto-currency transaction is performed. Must be populated when indicating crypto for VISA and MCC 6051. This is only applied to VISA OCT transactions. Contact Tech Support for more details. |
| card_holder | required | string(255) | Full name of customer as printed on credit card (first name and last name at least). Please, note that for **Visa** or **Master**, **Intl maestro** cards with gambling MCC **7995**, only Latin alphabet characters are accepted (**a..z**, **A..Z**) and any of: **. ' - 0..9**. Any other characters will be rejected and the transaction will be declined respectively. |
| card_number | required | 13 to 16 digits | Complete cc number of customer |
| cvv | required* | 3 to 4 digits | cvv of cc, requirement is based on terminal configuration |
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |
| token | optional | string(36) | See Tokenization for more details. If present, the cardholder parameters can be omitted. Cannot be set together with `remember_card` |
| remember_card | optional | "true" | See Tokenize. Tokenizes cardholder parameters. Cannot be set together with `token` |
| consumer_id | optional | string(10) | See Consumers and Tokenization. Combine with `remember_card` to tokenize or with `token` to use token |
| source_of_funds | optional | string | Specify the source of funds with one of `credit`, `debit`, `prepaid`, `cash`, `other_debit_account`, `other_credit_account`. |
| purpose_of_payment | required* | string (12) | Purpose of Payment code, required for Visa OCTs with recipients in Argentina, Bangladesh, Egypt, India, Chile and Colombia. |
| credential_on_file_transaction_identifier | optional | string(15) | The ID of the Account Funding Transactions, for money transfer OCTs. |
| **credential_on_file** | required* | | See Credential On File (COF) for more details |
| initial_customer_initiated | required* | string(18) | Initial transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | required* | string(18) | Subsequent transaction initiated by customer. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | optional | string(20) | Transaction is initiated by the merchant |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| document_id | required* | string(255) | Document ID value. |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |

| | | | |
|---|---|---|---|
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **customer_identification** | required* | | See Customer Identification Parameters for more details. |
| owner | required* | string(255) | The owner of the document ID |
| type | required* | string(255) | The type of the document ID |
| subtype | required* | string(255) | The subtype of the document ID |
| document_id | required* | string(255) | Document ID value. |
| issuing_country | required* | string(2) | The issuing country of the document ID |
| **account_owner** | optional | | Account owner parameters related to account owner inquiry requests |
| first_name | optional | string(35) | Account owner first name |
| middle_name | optional | string(35) | Account owner middle name |
| last_name | optional | string(35) | Account owner last name |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>payout</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>345678</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:45Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_response_code>00</scheme_response_code>
  <account_owner>
    <first_name>Travis</first_name>
    <middle_name>Joe</middle_name>
    <last_name>Pastrana</last_name>
  </account_owner>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenization |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |
| scheme_response_code | string(2) | The response code returned from the schemes. See Scheme response codes for details. |
| recurring_advice_code | string(2) | Additional response code returned from the schemes. See Recurring advice details |
| recurring_advice_text | string(255) | The text representation of the recurring advice code. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>payout</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <response_code>57</response_code>
  <code>340</code>
  <technical_message>billing_address[zip_code] is invalid!</technical_message>
  <message>billing_address[zip_code] is invalid!</message>
  <timestamp>2025-02-26T16:01:45Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### MONEY TRANSFER PAYOUT

Money transfer payout is a standard payout with additional parameters. The section and parameters below are optional and to be considered only when present.

Money transfers: account_to_account, person_to_person, wallet_transfer, funds_transfer, liquid_assets.

The transaction is not a result of a business operation but rather a pure funds movement from one account (card or non-card) to another (card).

Bear in mind that the sender of the funds in this case is not a merchant, but a consumer.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
```

```
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>payout</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4200000000000000</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <payment_transaction>service_provider_name</payment_transaction>
  <money_transfer>
    <type>account_to_account</type>
    <sender_account_number>DE91100000000123456789</sender_account_number>
    <sender_birth_date>24-09-1987</sender_birth_date>
    <service_provider_name>eMPPay</service_provider_name>
    <sender_address>
      <first_name>John</first_name>
      <last_name>Smith</last_name>
      <country>JoDEhn</country>
      <city>Berlin</city>
      <zip_code>10115</zip_code>
      <address1>1 Kaiserdamm Blvd, Berlin</address1>
    </sender_address>
  </money_transfer>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **money_transfer** | optional | | Money transfer Parameters |
| type | required | string | The type of money transfer. It can be account_to_account, person_to_person, wallet_transfer, funds_transfer, liquid_assets |
| sender_account_number | required | string(33) | Sender account number |
| sender_birth_date | optional | dd-mm-yyyy | Must contain valid birth date of the sender. |
| service_provider_name | optional | string(25) | Must contain a valid Service Provider Name. Only alphanumeric characters are allowed (including spaces). |
| **sender_address** | required | | |
| first_name | required | string(255) | Sender first name |
| last_name | required | string(255) | Sender last name |
| country | required | string(2) | Sender Country code in ISO 3166 |
| city | required | string(255) | Sender City |
| zip_code | required | string | Sender ZIP code |
| address1 | required | string(255) | Sender Primary address |
| state | required* | string(2) | Sender State code in ISO 3166-2, required for USA and Canada |

`required* = conditionally required`

The Sender name is split in two fields 'first name' and 'last name'. Maximum length of these fields must be 24 characters. Ex. 'John Doe' with space between them must not exceed 25 characters.

Money transfer is supported only by Visa.

| Money transfer type |
|---|
| account_to_account |
| person_to_person |
| wallet_transfer |
| funds_transfer |
| liquid_assets |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>payout</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <payment_response>response_code</payment_response>
  <timestamp>2025-02-26T16:01:45Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
</payment_response>
```

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>payout</transaction_type>
  <status>error</status>
  <mode>live</mode>
```

```
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>money_transfer_type is not supported. Please select one of account_to_account, person_to_person, wallet_transfer, funds_transfer, liquid_assets</technical_message>
  <message>Please check input data for errors!</message>
  <timestamp>2025-02-26T16:01:45Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
</payment_response>
```

**NON-MONEY TRANSFER PAYOUT**

Non-money transfer payout is a standard payout with additional parameters. The section and parameters below are optional and to be considered only when present.

Non-money transfer types are: b2b_supplier, loyalty, funds_disbursment, merchant_settlement, prepaid_card_load.

The transaction is a result of a business operation.

Bear in mind that the sender of the funds in this case is a merchant.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>payout</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <money_transfer>
     <type>loyalty</type>
   </money_transfer>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **money_transfer** | optional | | Non-money transfer Parameters |
| type | required | string | The type of non-money transfer. It can be b2b_supplier, loyalty, funds_disbursment, merchant_settlement, prepaid_card_load |

`required* = conditionally required`

Non-money transfer is supported only by Visa.

The listed non-money transfer types are allowed for all merchant category codes (MCCs).

| Non-money transfer type |
|---|
| b2b_supplier |
| loyalty |
| funds_disbursment |
| merchant_settlement |
| prepaid_card_load |

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>payout</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <payment_response>response_code</payment_response>
   <timestamp>2025-02-26T16:01:45Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
 </payment_response>
```

Error Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>payout</transaction_type>
   <status>error</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <code>340</code>
   <technical_message>money_transfer_type is not supported. Please select one of b2b_supplier, loyalty, funds_disbursment, merchant_settlement, prepaid_card_load</technical_message>
   <message>Please check input data for errors!</message>
   <timestamp>2025-02-26T16:01:45Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
 </payment_response>
```

**SCT PAYOUT**

SCT payouts are Sepa-based payouts to consumers done without a reference transaction and as such are regulated and need specific gateway terminal configuration, so be sure to contact the IT Support team at tech-support@e-comprocessing.com) in case you want SCT payouts to be enabled.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sct_payout</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>EUR</currency>
   <iban>DE09100100101234567891</iban>
   <bic>PBNKDEFFXXX</bic>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <country>DE</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| iban | required | string(34) | Customer’s IBAN number |
| bic | required | string(11) | SWIFT/BIC code of the customer’s bank |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required | string(255) | Customer first name |
| last_name | required | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **sct_payout** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| iban | required | string(34) | Customer’s IBAN number |
| bic | required | string(11) | SWIFT/BIC code of the customer’s bank |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required | string(255) | Customer first name |
| last_name | required | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |

| | | | |
|---|---|---|---|
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries:**

Allowed countries include supported countries for SDD Init Recurring Sale and the list below:

| Country Name | Country Code |
|---|---|
| Bulgaria | BG |
| Croatia | HR |
| Czech Republics | CZ |
| Denmark | DK |
| United Kingdom | UK |
| Hungary | HU |
| Iceland | IS |
| Liechtenstein | LI |
| Norway | NO |
| Poland | PL |
| Romania | RO |
| Sweden | SE |
| Switzerland | CH |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sct_payout</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:45Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sct_payout</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
```

```
    <code>340</code>
    <technical_message>expiration_year is invalid</technical_message>
    <message>expiration_year is invalid</message>
    <timestamp>2025-02-26T16:01:45Z</timestamp>
    <descriptor>Descriptor one</descriptor>
    <amount>100</amount>
    <currency>EUR</currency>
    <sent_to_acquirer>false</sent_to_acquirer>
  </payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### AFRICAN MOBILE PAYOUT

African Mobile Payout, or otherwise known as Disbursement, is an APM used to process Mobile network operator payments. It is an async payment method and will be approved once the payment is processed with the Mobile network operator

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>african_mobile_payout</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>KES</currency>
   <customer_email>barney.rubble@example.com</customer_email>
   <customer_phone>254701123456</customer_phone>
   <operator>SAFARICOM</operator>
   <target>000010</target>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Nairobi</city>
     <country>KE</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **african_mobile_payout** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| operator | required | string(20) | Name of the Mobile network operator (MNO) which should process the transaction |
| target | required | string(20) | Number of the Paybill for which the transaction is intended |
| customer_phone | required | string(32) | Must contain valid phone number of customer |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |

| | | | |
|---|---|---|---|
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries, currencies, operators and payment method:**

| Country code | Currency code | Operator | Payment method |
|---|---|---|---|
| GH | GHS | AIRTEL | Airtel Money |
| GH | GHS | MTN | MTN Mobile Money |
| GH | GHS | TIGO | Tigo Cash |
| GH | GHS | VODACOM | M-PESA |
| KE | KES | AIRTEL | Airtel Money |
| KE | KES | SAFARICOM | M-PESA |
| MZ | MZN | MOVITEL | e-Mola |
| MZ | MZN | VODACOM | M-PESA |
| RW | RWF | MTN | MTN Mobile Money |
| RW | RWF | TIGO | Tigo Cash |
| TZ | TZS | AIRTEL | Airtel Money |
| TZ | TZS | TIGO | Tigo Cash |
| TZ | TZS | VODACOM | M-PESA |
| UG | UGX | AIRTEL | Airtel Money |
| UG | UGX | MTN | MTN Mobile Money |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>african_mobile_payout</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:45Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>KES</currency>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>african_mobile_payout</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
```

```
    <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
    <message>Something went wrong, please contact support!</message>
    <technical_message>operator is not supported!</technical_message>
    <timestamp>2025-02-26T16:01:45Z</timestamp>
    <descriptor>Descriptor one</descriptor>
    <amount>100</amount>
    <currency>KES</currency>
  </payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| message | string(255) | Human readable error message which can be displayed to users. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

#### RUSSIAN MOBILE PAYOUT

Russian Mobile Payout, or otherwise known as Disbursement, is an APM used to process Mobile network operator payments. It is an async payment method and will be approved once the payment is processed by the Mobile network operator. Notice: Russian Mobile Payout does not support refund and void.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>russian_mobile_payout</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>RUB</currency>
   <customer_email>barney.rubble@example.com</customer_email>
   <customer_phone>79031234567</customer_phone>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Nairobi</city>
     <country>RU</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **russian_mobile_payout** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_phone | required | string(32) | Must contain valid phone number of customer |
| customer_email | optional | e-mail address | Must contain valid e-mail of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required | string(255) | Customer first name |
| last_name | required | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |

| | | | |
|---|---|---|---|
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries and currencies:**

| Country code | Currency code |
|---|---|
| RU | RUB |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>russian_mobile_payout</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:45Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>RUB</currency>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

#### TRANSFERTO PAYOUT

TransferTo Payout is an APM which provides 3 different payment services: BankAccount, MobileWallet and CashPickup. Merchant sends money to a consumer. Money are delivered through a Payer institution which supports one of the 3 services and has specific requirements on the transaction's amount and required fields. The process is async and once the TransferTo processes the transaction, a notification is received and the status is updated.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>transfer_to_payout</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>Funding</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <amount>100</amount>
  <currency>USD</currency>
  <customer_email>barney.rubble@example.com</customer_email>
  <payer_id>7</payer_id>
  <bank_account_number>0842024000</bank_account_number>
  <billing_address>
    <last_name>Rubble</last_name>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **transfer_to_payout** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |

| | | | |
|---|---|---|---|
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| payer_id | required | string | ID of the Payer used to deliver the money through one of the 3 services. For more details regarding how the available payers are populated see Retrieve Payers API |
| bank_account_number | required* | string | Bank identification number of the customer. *Requirement based on the Payer |
| indian_financial_system_code | required* | string | Bank code of the bank in which the consumer resides. *Requirement based on the Payer |
| msisdn | required* | string | Phone number for payment to bank account and wallet registration number for payment to wallet. Min 6 and max 32 digits. Numeric values only (can contain “+” at start or “(“ , ”)” , ”-“). *Requirement based on the Payer |
| branch_number | required* | string | Branch number. *Requirement based on the Payer |
| account_type | required* | string | Account type. Allowed values: CHECKING, SAVINGS, DEPOSIT, OTHERS. *Requirement based on the Payer |
| registered_name | required* | string | Registered name of the business. *Requirement based on the Payer |
| registration_number | required* | string | Registration number. *Requirement based on the Payer |
| document_reference_number | required* | string | Reference number for the contract. *Requirement only for Business-to-Business (B2B) workflow |
| purpose_of_remittance | required* | string | Identification type. Allowed values: FAMILY_SUPPORT, EDUCATION, GIFT_AND_DONATION, MEDICAL_TREATMENT, MAINTENANCE_EXPENSES, TRAVEL, SMALL_VALUE_REMITTANCE, LIBERALIZED_REMITTANCE, OTHER. *Requirement only for Business-to-Business (B2B) workflow |
| iban | required* | string | Bank account number in IBAN format. *Requirement based on the Payer |
| id_type | required* | string | Identification type. Allowed values: PASSPORT, NATIONAL_ID, DRIVING_LICENSE, SOCIAL_SECURITY, TAX_ID, SENIOR_CITIZEN_ID, BIRTH_CERTIFICATE, VILLAGE_ELDER_ID, RESIDENT_CARD, ALIEN_REGISTRATION, PAN_CARD, VOTERS_ID, HEALTH_CARD, EMPLOYER_ID, OTHER. *Requirement based on the Payer |
| id_number | required* | string | Identification number. *Requirement based on the Payer |
| sender_date_of_birth | required* | string | Date of birth with the following format YYYY-MM-DD. *Requirement based on the Payer |
| sender_last_name | required* | string | First name of the sender. *Requirement based on the Payer |
| sender_first_name | required* | string | First name of the sender. *Requirement based on the Payer |
| sender_country_iso_code | required* | string | Three-letter country code of the sender. *Requirement based on the Payer |
| sender_id_number | required* | string | Identification number of the sender. *Requirement based on the Payer |
| sender_nationality_country_iso_code | required* | string | Three-letter country code corresponding to the nationality of the sender. *Requirement based on the Payer |
| sender_address | required* | string | First line of address of the sender. *Requirement based on the Payer |
| sender_occupation | required* | string | Occupation of the sender. *Requirement based on the Payer |
| sender_beneficiary_relationship | required* | string | Relationship between the sender and the beneficiary. *Requirement based on the Payer |
| sender_postal_code | required* | string | Postal code of the sender. *Requirement based on the Payer |
| sender_city | required* | string | City of the sender. *Requirement based on the Payer |
| sender_msisdn | required* | string | Phone number for payment to bank account and wallet registration number for payment to wallet. Min 6 and max 32 digits. Numeric values only (can contain “+” at start or “(“ , ”)” , ”-“). *Requirement based on the Payer |
| sender_gender | required* | string | Gender of the sender. *Requirement based on the Payer |
| sender_id_type | required* | string | Identification type of the sender. Allowed values: PASSPORT, NATIONAL_ID, DRIVING_LICENSE, SOCIAL_SECURITY, TAX_ID, SENIOR_CITIZEN_ID, BIRTH_CERTIFICATE, VILLAGE_ELDER_ID, RESIDENT_CARD, ALIEN_REGISTRATION, PAN_CARD, VOTERS_ID, HEALTH_CARD, EMPLOYER_ID, OTHER. *Requirement based on the Payer |
| sender_province_state | required* | string | Province State of the sender. *Requirement based on the Payer |
| sender_source_of_funds | required* | string | Source of funds of the sender. *Requirement based on the Payer |
| sender_country_of_birth_iso_code | required* | string | Three-letter country code corresponding to the country of birth of the sender. *Requirement based on the Payer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(3) | Three-letter country code in alpha-3 format |

`required* = conditionally required`

**Supported currencies:**

| Currency |
|---|
| EUR |
| GBP |
| HKD |
| USD |

**Supported destination countries and currencies:**

| Country | Country Code | Currency |
|---|---|---|
| Brazil | BRA | BRL |
| China | CHN | CNY |
| Indonesia | IDN | IDR |

| | | |
|---|---|---|
| Malaysia | MYS | MYR |
| Philippines | PHL | PHP |
| Thailand | THA | THB |
| Argentina | ARG | ARS |
| Chile | CHL | CLP |
| Mexico | MEX | MXN |
| Peru | PER | PEN |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>transfer_to_payout</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:45Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>transfer_to_payout</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>910</code>
  <technical_message>Payer is currently unavailable</technical_message>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:45Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

---

# Mobile Payments

**APPLE PAY**

Apple Pay is a mobile payment solution available on iOS devices with Touch ID / Face ID support. Apple Pay allows shoppers to purchase with credit and debit cards linked to their devices.

The Apple Pay is supported on the Web Payment Form via Express Checkout.

Payment Object structure

```
{
  "token": {
    "paymentData": { ... },
    "paymentMethod": { ... },
    "transactionIdentifier": "32b...4f3"
  },
  "billingContact": { ... },
  "shippingContact": { ... }
}
```

To use the Apple Pay, your application should be set up with public, private keys and payment processing certificate. Please contact tech-support@e-comprocessing.com to enable Apple Pay payments.

Once a payment authorized by the customer in the merchant's application, the Apple Pay APIs will return a Payment Object containing the payment data (Encrypted Payment Token) with customer information to the merchant's application. On the right is an example of the Payment Object structure that will be returned to the merchant's application.

Once a Payment Object received, it can be used to create an Apple Pay payment transaction. To create an Apple Pay payment transaction you should specify the Encrypted Payment Token in the `<payment_token>` tag and the payment type you need in the `<payment_subtype>` tag. To specify Encrypted Payment Token use the value of the `token` field of the received Payment Object.

The following payment types are supported:

| Payment type | Description |
|---|---|
| authorize | behaves like common authorize transaction |
| init_recurring_sale | behaves like common init_recurring_sale transaction |
| sale | behaves like common sale transaction |

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>apple_pay</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_subtype>authorize</payment_subtype>
   <recurring_type>initial</recurring_type>
   <payment_token>
     {
       "paymentData": { ... },
       "paymentMethod": { ... },
       "transactionIdentifier": "32B...4F3"
     }
   </payment_token>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **apple_pay** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| payment_token | required | | Encrypted Payment Token |
| payment_subtype | required | | Use either **authorize** for Authorize or **sale** for Sale transactions. |
| usage | optional | string(255) | Description of the transaction for later use. |
| recurring_type | optional | string(255) | Specifies recurring type of the transaction. Use **initial**. |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount to act as an account verification transaction - Contact tech-support@e-comprocessing.com for more details regarding this scenario. |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| document_id | required* | string(255) | Document ID value. |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |

| | | | |
|---|---|---|---|
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |
| **sender** | optional | | Funding Transaction Sender details |
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |

| | | | |
|---|---|---|---|
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>apple_pay</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
</payment_response>
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>apple_pay</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <payment_subtype>authorize</payment_subtype>
  <payment_token>
    {
      "paymentData": { ... },
      "paymentMethod": { ... },
      "transactionIdentifier": "32B...4F3"
    }
  </payment_token>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>0</amount>
  <currency>USD</currency>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <business_attributes>
    <event_start_date>27-03-2025</event_start_date>
    <event_end_date>08-04-2025</event_end_date>
    <event_organizer_id>20192375</event_organizer_id>
    <event_id>1912</event_id>
  </business_attributes>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>apple_pay</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>0</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |

| | | |
|---|---|---|
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to 'true' if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| payment_account_reference | string(255) | Payment Account Reference value returned from the schemes. Links tokenized and PAN-based transactions. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>apple_pay</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>expiration_year is invalid</technical_message>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**GOOGLE PAY**

Google Pay allows shoppers to purchase with credit and debit cards linked to their Google account.

Strong Customer Authentication thru 3DS might be required in case the cardholder has not yet been authenticated previously and the transaction is **PAN_ONLY**.
In certain cases, a 3DS challenge may be required or 3DS-Method to be submitted first, it is highly recommended to submit 3DSv2 params.

The Google Pay is supported on the Web Payment Form via Express Checkout.

Payment Object structure

```
{
  "signature":"...",
  "intermediateSigningKey":{
    "signedKey":"{
      "keyValue": "...",
      "keyExpiration": "..."
    }",
    "signatures":[...]
  },
  "protocolVersion":"...",
  "signedMessage":"{
    "encryptedMessage": "...",
    "ephemeralPublicKey": "...",
    "tag": "..."
  }"
}
```

Once payment is authorized by the customer in the merchant's application, the Google Pay APIs will return a Payment Method Token containing the payment data (Encrypted Payment Token) with customer information to the

merchant’s application. On the right is an example of the Payment Method Token structure that will be returned to the merchant’s application.

Once a Payment Object is received, it can be used to create a Google Pay payment transaction. To create a Google Pay payment transaction you should specify the Encrypted Payment Token in the `<payment_token>` tag and the payment type you need in the `<payment_subtype>` tag.

The following payment types are supported:

| Payment type | Description |
|---|---|
| authorize | behaves like common authorize transaction |
| init_recurring_sale | behaves like common init_recurring_sale transaction |
| sale | behaves like common sale transaction |

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>google_pay</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_subtype>authorize</payment_subtype>
   <recurring_type>initial</recurring_type>
   <payment_token>
     {
       "signature":"...",
       "intermediateSigningKey":{
         "signedKey":"{
           "keyValue": "...",
           "keyExpiration": "..."
         }",
         "signatures":[...]
       },
       "protocolVersion":"...",
       "signedMessage":"{
         "encryptedMessage": "...",
         "ephemeralPublicKey": "...",
         "tag": "..."
       }"
     }
   </payment_token>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

Asynchronous 3 D Sv2 Challenge With 3 Ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>google_pay</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_subtype>authorize</payment_subtype>
   <payment_token>
     {
       "signature":"...",
       "intermediateSigningKey":{
         "signedKey":"{
           "keyValue": "...",
           "keyExpiration": "..."
         }",
         "signatures":[...]
       },
       "protocolVersion":"...",
       "signedMessage":"{
         "encryptedMessage": "...",
         "ephemeralPublicKey": "...",
         "tag": "..."
       }"
     }
   </payment_token>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
```

```
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <recurring>
      <expiration_date>27-08-2025</expiration_date>
      <frequency>30</frequency>
    </recurring>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
</payment_transaction>'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>google_pay</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <payment_subtype>authorize</payment_subtype>
  <payment_token>
    {
      "signature":"...",
      "intermediateSigningKey":{
        "signedKey":"{
          "keyValue": "...",
          "keyExpiration": "..."
        }",
        "signatures":[...]
      },
      "protocolVersion":"...",
      "signedMessage":"{
        "encryptedMessage": "...",
        "ephemeralPublicKey": "...",
        "tag": "..."
      }"
    }
  </payment_token>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>0</amount>
  <currency>USD</currency>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <business_attributes>
    <event_start_date>27-03-2025</event_start_date>
    <event_end_date>08-04-2025</event_end_date>
    <event_organizer_id>20192375</event_organizer_id>
    <event_id>1912</event_id>
  </business_attributes>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **google_pay** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |

| | | | |
|---|---|---|---|
| payment_token | required | | Encrypted Payment Token |
| payment_subtype | required | | Use either **authorize** for Authorize or **sale** for Sale transactions. |
| recurring_type | optional | string(255) | Specifies recurring type of the transaction. Use **initial**. |
| usage | optional | string(255) | Description of the transaction for later use. |
| amount | required | integer >= 0 | Transaction amount in minor currency unit, see Currency and Amount Handling for details. In certain cases, it is possible to submit a transaction with a zero-value amount to act as an account verification transaction - Contact tech-support@e-comprocessing.com for more details regarding this scenario. |
| currency | required | string(3) | Currency code in ISO 4217 |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| birth_date | required* | dd-mm-yyyy | Required when MCC is a Financial Services one (e.g. MCC 6012) and either card brand is Visa or Mastercard/Maestro with UK-based merchant, UK-based bin (domestic), and DEBIT card type |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| document_id | required* | string(255) | Document ID value. |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| notification_url | required* | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required* | url | URL where customer is sent to after successful payment |
| return_failure_url | required* | url | URL where customer is sent to after unsuccessful payment |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |
| **sender** | optional | | Funding Transaction Sender details |
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |

| | | | |
|---|---|---|---|
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |
| **threeds_v2_params** | required* | | 3DSv2 async parameters. They must be submitted in order to use the 3DSv2 authentication protocol in asynchronous workflow |
| **threeds_method** | optional | | 3DS-Method related parameters for any callbacks and notifications. |
| callback_url | optional | url | Specific 3DS-Method callback URL after the 3DS-Method completes. The actual status will be provided via HTTP POST to that URL. For more information, go to 3DSv2 method params |
| **control** | required* | | General params for preferences in authentication flow and providing device interface information. |
| device_type | required* | string | Identifies the device channel of the consumer, **required** in the 3DSv2 authentication protocol. For more information, go to 3DSv2 control params |
| challenge_window_size | required* | string | Identifies the size of the challenge window for the consumer. For more information, go to 3DSv2 control params |
| challenge_indicator | optional | string | The value has weight and might impact the decision whether a challenge will be required for the transaction or not. If not provided, it will be interpreted as **no_preference**. For more information, go to 3DSv2 control params |
| **purchase** | optional | | Purchase related params providing with additional information regarding the order. |
| category | optional | string | Identifies the type of transaction being authenticated. This field is required in some markets.Accepted values are: **goods**, **service**, **check_acceptance**, **account_funding**, **quasi_cash**, **prepaid_activation**, **loan**. |
| **merchant_risk** | recommended | | Merchant risk assessment params. They are all optional, but recommended. |
| shipping_indicator | optional | string(16) | Indicator code that most accurately describes the shipping method for the cardholder specific transaction. If one or more items are included in the sale, use the Shipping Indicator code for the physical goods. If all digital goods, use the code that describes the most expensive item. Accepted values are: **same_as_billing**, **stored_address**, **verified_address**, **pick_up**, **digital_goods**, **travel**, **event_tickets**, **other**. |
| delivery_timeframe | optional | string(11) | Indicates the merchandise delivery timeframe. Accepted values are: **electronic**, **same_day**, **over_night**, **another_day**. |
| reorder_items_indicator | optional | string(10) | Indicates whether the cardholder is reordering previously purchased merchandise. Accepted values are: **first_time**, **reordered**. |
| pre_order_purchase_indicator | optional | string(21) | Indicates whether cardholder is placing an order for merchandise with a future-availability or release date. Accepted values are: **merchandise_available**, **future_availability**. |
| pre_order_date | optional | dd-mm-yyyy | For a pre-ordered purchase, the expected date that the merchandise will be available. |
| gift_card | optional | 'true' | Prepaid or gift card purchase. |
| gift_card_count | optional | integer | For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased. The value is limited to **99**. |
| **card_holder_account** | recommended | | Cardholder account additional information. They are all optional, but recommended, because they have a significant impact on approval rates |
| creation_date | optional | dd-mm-yyyy | Date that the cardholder opened the account with the 3DS Requester. |
| update_indicator | optional | string(19) | Length of time since the cardholder’s account information with the 3DS Requestor was last changed. Includes Billing or Shipping address, new payment account, or new user(s) added. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| last_change_date | optional | dd-mm-yyyy | Date that the cardholder’s account with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment accout, or new user(s) added. |
| password_change_indicator | optional | string(18) | Length of time since the cardholder account with the 3DS Requestor had a password change or account reset. Accepted values are: **no_change**, **during_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| password_change_date | optional | dd-mm-yyyy | Date that cardholder’s account with the 3DS Requestor had a password change or account reset. |
| shipping_address_usage_indicator | optional | string(19) | Indicates when the shipping address used for this transaction was first used with the 3DS Requestor. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| shipping_address_date_first_used | optional | dd-mm-yyyy | Date when the shipping address used for this transaction was first used with the 3DS Requestor. |
| transactions_activity_last_24_hours | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours. |
| transactions_activity_previous_year | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year. |
| provision_attempts_last_24_hours | optional | integer | Number of Add Card attempts in the last 24 hours. |
| purchases_count_last_6_months | optional | integer | Number of purchases with this cardholder account during the previous six months. |
| suspicious_activity_indicator | optional | string(22) | Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account. Accepted values are: **no_suspicious_observed**, **suspicious_observed**. |
| registration_indicator | optional | string(19) | Indicates the length of time that the payment account was enrolled in the cardholder’s account with the 3DS Requester. Accepted values are: **guest_checkout**, **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| registration_date | optional | dd-mm-yyyy | Date that the payment account was enrolled in the cardholder’s account with the 3DS Requestor. |
| **browser** | required* | | For browser-based transactions. They are all *required* in case the **device_type** is set to **browser** |
| accept_header | required* | string(2048) | The exact content of the HTTP **ACCEPT** header as sent to the 3DS Requester from the Cardholder browser. Any other header different than the **ACCEPT** header will be rejected. Example: `application/json,text/plain,text/html,*/*`. |
| java_enabled | required* | boolean | Boolean that represents the ability of the cardholder browser to execute Java. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.javaEnabled`. |
| language | required* | string(8) | Value representing the browser language as defined in IETF BCP47. Note that only one browser language tag is about to be submitted as per the above **IETF BCP47**. Numeric chars are also allowed in the subtag and will represent the region.Example: `en-GB`, `zh-guoyu`, `fil-PH`, `gsw`, `es-419`, `de-1996`, etc. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.language`. |
| color_depth | required* | integer | Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from Cardholder browser using the `screen.colorDepth` property. The value as per EMVCo specs can be one of **1**, **4**, **8**, **15**, **16**, **24**, **32**, **48**. In case, an unsupported `color_depth` is determined, the nearest supported value that is less than the actual one needs to be submitted. For example, if the obtained value is **30**, which is not supported as per EMVCo specs, **24** has to be submitted. |
| screen_height | required* | integer | Total height of the Cardholder's screen in pixels. Value is returned from the screen.height property. |
| screen_width | required* | integer | Total width of the Cardholder's screen in pixels. Value is returned from the screen.width property. |
| time_zone_offset | required* | string(5) | Time difference between UTC time and the Cardholder browser local time, in **minutes**. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. If **UTC -5** hours then submit `300` or `+300`, If **UTC +2** hours then `-120`. The value can be retrieved using Javascript `getTimezoneOffset()` method over **Date** object. |
| user_agent | required* | string(2048) | Exact content of the HTTP user-agent header. |
| **sdk** | required* | | For application-based transactions. They are all *required* in case the **device_type** is set to **application** |
| interface | required* | string(6) | SDK Interface types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. Accepted values are: **native**, **html**, **both**. |
| **ui_types** | required* | | Lists all UI types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. |

| | | | |
|---|---|---|---|
| ui_type | required* | string(13) | UI type that the device of the consumer supports for displaying specific challenge interface. Accepted values are: **text**, **single_select**, **multi_select**, **out_of_bag**, **other_html**. |
| application_id | required* | string(36) | Universally unique ID created upon all installations and updates of the 3DS Requestor APp on a Customer Device. This will be newly generated and stored by the 3DS SDK for each installation or update. The field is limited to 36 characters and it shall have a canonical format as defined in IETF RFC 4122. |
| encrypted_data | required* | string(64000) | JWE Object as defined Section 6.2.2.1 containing data encrypted by the SDK for the DS to decrypt. The data will be present when sending to DS, but not present from DS to ACS. |
| ephemeral_public_key_pair | required* | string(256) | Public key component of the ephemeral key pair generated by the 3DS SDK and used to establish session keys between the 3DS SDK and ACS. In AReq, this data element is contained within the ACS Signed Content JWS Object. The field is limited to maximum 256 characters. |
| max_timeout | required* | integer | Indicates the maximum amount of time (in minutes) for all exchanges. The field shall have value greater or equals than 05. |
| reference_number | required* | string(32) | Identifies the vendor and version of the 3DS SDK that is integrated in a 3DS Requestor App, assigned by EMVCo when the 3DS SDK is approved. The field is limited to 32 characters. |
| **recurring** | optional | | Additional recurring details. |
| expiration_date | optional | dd-mm-yyyy | A future date indicating the end date for any further subsequent transactions. For more information, go to 3DSv2 recurring params |
| frequency | optional | integer | Indicates the minimum number of days between subsequent transactions. An empty value indicates the payment frequency is not set.For more information, go to 3DSv2 recurring params |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>google_pay</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
</payment_response>
```


Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>google_pay</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>0</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
</payment_response>
```


Challenge Without 3 Ds Method Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>google_pay</transaction_type>
  <status>pending_async</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
```

```
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <redirect_url>https://staging.gate.e-comprocessing.net/threeds/authentication/44177a21403427eb96664a6d7e5d5d48</redirect_url>
  <redirect_url_type>3ds_v2_challenge</redirect_url_type>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```


Asynchronous 3 D Sv2 Challenge With 3 Ds Method Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>google_pay</transaction_type>
  <status>pending_async</status>
  <mode>test</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <threeds_method_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method</threeds_method_url>
  <threeds_method_continue_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method/44177a21403427eb96664a6d7e5d5d48</threeds_method_continue_url>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| redirect_url | url | URL where user has to be redirected to complete payment process. It is available for asynchronous mode |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ‘true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |
| scheme_transaction_identifier | string(32) | Id defined by card schemes. Corresponds to NETWORK DATA (field 63) for MasterCard or TRANS ID (field 62.2/125) for VISA. |
| payment_account_reference | string(255) | Payment Account Reference value returned from the schemes. Links tokenized and PAN-based transactions. |

Error Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>google_pay</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>340</code>
  <technical_message>expiration_year is invalid</technical_message>
  <message>expiration_year is invalid</message>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```


**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |

| | | |
|---|---|---|
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

**Supported countries**

| Country Name | Country Code |
|---|---|
| Albania | AL |
| Algeria | DZ |
| American Samoa | AS |
| Andorra | AD |
| Angola | AO |
| Antigua and Barbuda | AG |
| Argentina | AR |
| Armenia | AM |
| Australia | AU |
| Austria | AT |
| Azerbaijan | AZ |
| Bahamas | BS |
| Bahrain | BH |
| Bangladesh | BD |
| Barbados | BB |
| Belarus | BY |
| Belgium | BE |
| Belize | BZ |
| Benin | BJ |
| Bhutan | BT |
| Bolivia | BO |
| Bosnia and Herzegovina | BA |
| Botswana | BW |
| Brazil | BR |
| Brunei | BN |
| Bulgaria | BG |
| Burkina Faso | BF |
| Burundi | BI |
| Cambodia | KH |
| Cameroon | CM |
| Canada | CA |
| Cape Verde | CV |
| Central African Republic | CF |
| Chad | TD |
| Chile | CL |
| China | CN |
| Colombia | CO |
| Comoros | KM |
| Congo (Democratic Republic) | CD |
| Congo (Republic) | CG |
| Costa Rica | CR |
| Croatia | HR |
| Cyprus | CY |
| Czech Republic | CZ |
| Czechia | CZ |
| Denmark | DK |
| Djibouti | DJ |
| Dominica | DM |

| | |
|---|---|
| Dominican Republic | DO |
| Ecuador | EC |
| Egypt | EG |
| El Salvador | SV |
| Equatorial Guinea | GQ |
| Eritrea | ER |
| Estonia | EE |
| Eswatini | SZ |
| Ethiopia | ET |
| Fiji | FJ |
| Finland | FI |
| France | FR |
| Gabon | GA |
| Gambia | GM |
| Georgia | GE |
| Germany | DE |
| Ghana | GH |
| Greece | GR |
| Grenada | GD |
| Guatemala | GT |
| Guinea | GN |
| Guinea-Bissau | GW |
| Guyana | GY |
| Haiti | HT |
| Honduras | HN |
| Hong Kong | HK |
| Hungary | HU |
| Iceland | IS |
| India | IN |
| Indonesia | ID |
| Iran | IR |
| Iraq | IQ |
| Ireland | IE |
| Israel | IL |
| Italy | IT |
| Ivory Coast | CI |
| Jamaica | JM |
| Japan | JP |
| Jordan | JO |
| Kazakhstan | KZ |
| Kenya | KE |
| Kiribati | KI |
| Kuwait | KW |
| Kyrgyzstan | KG |
| Laos | LA |
| Latvia | LV |
| Lebanon | LB |
| Lesotho | LS |
| Liberia | LR |
| Libya | LY |
| Liechtenstein | LI |
| Lithuania | LT |
| Luxembourg | LU |
| Macao | MO |
| Macedonia | MK |
| Madagascar | MG |
| Malawi | MW |
| Malaysia | MY |
| Maldives | MV |

| | |
|---|---|
| Mali | ML |
| Malta | MT |
| Marshall Islands | MH |
| Mauritania | MR |
| Mauritius | MU |
| Mexico | MX |
| Micronesia | FM |
| Moldova | MD |
| Monaco | MC |
| Mongolia | MN |
| Montenegro | ME |
| Morocco | MA |
| Mozambique | MZ |
| Myanmar | MM |
| Namibia | NA |
| Nauru | NR |
| Nepal | NP |
| Netherlands | NL |
| New Zealand | NZ |
| Nicaragua | NI |
| Niger | NE |
| Nigeria | NG |
| Norway | NO |
| Oman | OM |
| Pakistan | PK |
| Palau | PW |
| Palestine | PS |
| Panama | PA |
| Papua New Guinea | PG |
| Paraguay | PY |
| Peru | PE |
| Philippines | PH |
| Poland | PL |
| Portugal | PT |
| Qatar | QA |
| Romania | RO |
| Russian Federation | RU |
| Rwanda | RW |
| Saint Kitts and Nevis | KN |
| Saint Lucia | LC |
| Saint Vincent and the Grenadines | VC |
| Samoa | WS |
| San Marino | SM |
| São Tomé and Príncipe | ST |
| Saudi Arabia | SA |
| Senegal | SN |
| Serbia | RS |
| Seychelles | SC |
| Sierra Leone | SL |
| Singapore | SG |
| Slovakia | SK |
| Slovenia | SI |
| Solomon Islands | SB |
| Somalia | SO |
| South Africa | ZA |
| South Korea | KR |
| Spain | ES |
| Sri Lanka | LK |
| Sudan | SD |
| Suriname | SR |

| | |
|---|---|
| Sweden | SE |
| Switzerland | CH |
| Syria | SY |
| Taiwan | TW |
| Tajikistan | TJ |
| Tanzania | TZ |
| Thailand | TH |
| Togo | TG |
| Tonga | TO |
| Trinidad and Tobago | TT |
| Tunisia | TN |
| Turkey | TR |
| Turkmenistan | TM |
| Uganda | UG |
| Ukraine | UA |
| United Arab Emirates | AE |
| United Kingdom | GB |
| United States Minor Outlying Islands | UM |
| United States of America | US |
| Uruguay | UY |
| Uzbekistan | UZ |
| Vanuatu | VU |
| Vatican City | VA |
| Venezuela | VE |
| Vietnam | VN |
| Yemen | YE |
| Zambia | ZM |
| Zimbabwe | ZW |

#### AFRICAN MOBILE SALE

African Mobile Sale, otherwise known as Charge, is an APM used to process Mobile network operator payments. It is an async payment method and will be approved once the payment is processed with the Mobile network operator

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>african_mobile_sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>KES</currency>
   <customer_email>barney.rubble@example.com</customer_email>
   <customer_phone>254701123456</customer_phone>
   <operator>SAFARICOM</operator>
   <target>000010</target>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Nairobi</city>
     <country>KE</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **african_mobile_sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| operator | required | string(20) | Name of the Mobile network operator (MNO) which should process the transaction |

| | | | |
|---|---|---|---|
| target | required | string(20) | Number of the Paybill for which the transaction is intended |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries, currencies, operators and payment method:**

| Country code | Currency code | Operator | Payment method |
|---|---|---|---|
| GH | GHS | VODACOM | M-PESA |
| KE | KES | SAFARICOM | M-PESA |
| UG | UGX | AIRTEL | Airtel Money |
| UG | UGX | MTN | MTN Mobile Money |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>african_mobile_sale</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>KES</currency>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>african_mobile_sale</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <message>Something went wrong, please contact support!</message>
  <technical_message>operator is not supported</technical_message>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>KES</currency>
```

```
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| message | string(255) | Human readable error message which can be displayed to users. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

#### RUSSIAN MOBILE SALE

Russian Mobile Sale, otherwise known as Charge, is an APM used to process Mobile network operator payments. It is an async payment method and will be approved once the payment is processed by the Mobile network operator. Notice: Russian Mobile Sale does not support refund and void.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>russian_mobile_sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>152</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <amount>100</amount>
   <currency>RUB</currency>
   <customer_email>barney.rubble@example.com</customer_email>
   <customer_phone>79031234567</customer_phone>
   <operator>megafon</operator>
   <target>15472</target>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Nairobi</city>
     <country>RU</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **russian_mobile_sale** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required | string(5) | Customer account number or order identificator in the merchant system. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| return_success_url | required | url | URL where customer is sent to after successful payment |
| return_failure_url | required | url | URL where customer is sent to after unsuccessful payment |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| operator | required | string | Mobile network operator name (**mtc**, **megafon**, **tele2** or **beeline**). |
| target | required | integer | Merchant prefix. Unique for each mobile network operator assigned for the merchant. |
| customer_email | optional | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required | string(32) | Must contain valid phone number of customer |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |

| | | | |
|---|---|---|---|
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

**Supported countries, currencies and operators:**

| Country code | Currency code | Operator |
|---|---|---|
| RU | RUB | MTC |
| RU | RUB | Megafon |
| RU | RUB | Tele2 |
| RU | RUB | Beeline |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>russian_mobile_sale</transaction_type>
  <status>pending_async</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>RUB</currency>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>russian_mobile_sale</transaction_type>
  <status>error</status>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <message>Something went wrong, please contact support!</message>
  <technical_message>operator is not supported</technical_message>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>RUB</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| message | string(255) | Human readable error message which can be displayed to users. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

## Reversals

Reversal transactions serve to change the state of the original transaction and return money back to customer’s account. They can be used with Card, 3DS Card, and different APM transactions.

**REFUND**

Refunds allow to return already billed amounts to customers.

The amount can be fully or partially refunded. Refunds can only be done as follow transactions on former successfully processed transactions:

- Card transactions
- 3DS Card transactions
- Wallets
- Vouchers
- Online Banking ePayments
- Cash payments
- Gift Cards

Therefore, the reference id for the corresponding transaction is mandatory.

This transaction type supports Level 3 travel data.

This transaction type supports Installments.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>refund</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
   <amount>100</amount>
   <currency>USD</currency>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **refund** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| beneficiary_bank_code | required* | string | The bank code of the beneficiary's bank. |
| beneficiary_name | required* | string | The name of the beneficiary's bank. |
| beneficiary_account_number | required* | string | The account number of the beneficiary in his bank. |
| bank | optional | string | Name of the customer's bank |
| bank_branch | optional | string | Name of the Bank branch |
| bank_account | optional | string | Bank account number of the customer. |
| bank_account_type | optional | string(1) | The type of account. C: for Checking accounts, S: for Savings accounts, I: for International accounts |

`required* = conditionally required`

Beneficiary params will be required when refunding an online banking transaction with MYR currency. Contact tech-support for more information.

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>refund</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <authorization_code>345678</authorization_code>
   <retrieval_reference_number>016813015184</retrieval_reference_number>
   <response_code>00</response_code>
   <timestamp>2025-02-26T16:01:46Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
 </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |

| | | |
|---|---|---|
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>refund</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>410</code>
  <technical_message>no approved reference transaction found</technical_message>
  <message>no approved reference transaction found</message>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

#### ASYNC REFUND

Async Refunds allow to return already billed amounts to customers where the transaction is confirmed asynchronously (i.e. the transaction is sent for processing without the client being blocked waiting and once the transaction is complete the client is notified about the result).

The amount can be fully or partially refunded. Async Refunds are only required for a few APMs. Please contact tech-support@e-comprocessing.com for more details.

Similarly to ordinary Refunds, the reference id for the corresponding transaction is mandatory.

This transaction type supports Level 3 travel data.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>async_refund</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
   <amount>100</amount>
   <currency>USD</currency>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **async_refund** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| beneficiary_bank_code | required* | string | The bank code of the beneficiary's bank. |
| beneficiary_name | required* | string | The name of the beneficiary's bank. |
| beneficiary_account_number | required* | string | The account number of the beneficiary in his bank. |
| bank | optional | string | Name of the customer's bank |
| bank_branch | optional | string | Name of the Bank branch |
| bank_account | optional | string | Bank account number of the customer. |

| | | | |
|---|---|---|---|
| bank_account_type | optional | string(1) | The type of account. C: for Checking accounts, S: for Savings accounts, I: for International accounts |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>async_refund</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>async_refund</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>410</code>
  <technical_message>no approved reference transaction found</technical_message>
  <message>no approved reference transaction found</message>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

#### VOID

Void transactions undo other transactions.

Transactions of types authorize, capture, sale, refund, recurring sale, and init recurring sale along with their 3D variants can be reversed on the same day the transaction took place. The transaction will not show up on the customer’s credit card statement if voided on the same day.

Not captured authorize and authorize3d transactions can be voided without a specific timeframe.

The same day is dependent of the timezone of the acquiring bank.

This transaction can also be used to fully reverse a Preauthorization The void time-frame in this case depends on the preauthorization specifics (Cardbrand, Merchant Category Code etc). To learn more about this, navigate to the Full Reversal section.

When reversing transaction using Void while it is in a process of settlement, an error `Transaction already scheduled for settlement!` will be returned.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>void</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **void** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>void</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <authorization_code>345678</authorization_code>
   <retrieval_reference_number>016813015184</retrieval_reference_number>
   <response_code>00</response_code>
   <timestamp>2025-02-26T16:01:46Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
   <sent_to_acquirer>true</sent_to_acquirer>
 </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>void</transaction_type>
   <status>error</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <code>420</code>
   <technical_message>can not do void on void reference</technical_message>
   <message>can not do void on void reference</message>
   <timestamp>2025-02-26T16:01:46Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
   <sent_to_acquirer>false</sent_to_acquirer>
 </payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |

| | | |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### INVOICE REFUND

**Invoice Refunds allow to return already billed amounts to customers.**

The amount can be fully or partially refunded. Invoice refunds can only be done on former Invoice Capture (settled) transactions.

Therefore, the `reference_id` for the corresponding transaction is mandatory.

> In case of **secure_invoice** payment type we can invoke purchase amount reduction of not yet captured invoice transaction by setting up **reference_id** from **InvoiceTransaction**. Then **items** section becomes items to remove from original invoice request.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>invoice_refund</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <payment_type>klarna</payment_type>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
   <amount>60</amount>
   <currency>EUR</currency>
   <items>
     <item>
       <item_type>physical</item_type>
       <reference>19-402-USA</reference>
       <name>BatteryPowerPack</name>
       <quantity>1</quantity>
       <unit_price>60</unit_price>
       <tax_rate>0</tax_rate>
       <total_amount>60</total_amount>
       <total_discount_amount>0</total_discount_amount>
       <total_tax_amount>0</total_tax_amount>
       <image_url>https://example.com/image_url</image_url>
       <product_url>https://example.com/product_url</product_url>
       <quantity_unit>pcs</quantity_unit>
       <merchant_data>
         <marketplace_seller_info>Electronic gadgets</marketplace_seller_info>
       </merchant_data>
     </item>
   </items>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **invoice_refund** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| payment_type | required | string | Payment provider type: **klarna** / **secure_invoice** |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | Unique id returned by corresponding transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| **items** | required | | List with items |
| item_type | required | string(255) | Order line type. Possible values: Supported item types |
| quantity | required | integer | Non-negative. The item quantity |
| unit_price | required | integer | Minor units. Includes tax, excludes discount(max value: 100000000) |
| total_amount | required | integer | Includes tax and discount. Must match (quantity unit price) - total discount amount divided by quantity (max value: 100000000) |
| reference | optional | string(255) | Article number, SKU or similar |
| name | optional | string(255) | Descriptive item name |
| tax_rate | optional | integer | Non-negative. In percent, two implicit decimals. I.e 2500 = 25.00 percent |
| total_discount_amount | optional | integer | Non-negative minor units. Includes tax |
| total_tax_amount | optional | integer | Must be within 1 of total amount - total_amount * 10000 / (10000 + tax rate). Negative when type is discount |
| image_url | optional | url | URL to an image that can be later embedded in communications between Klarna and the customer. (max 1024 characters) |
| product_url | optional | url | URL to an image that can be later embedded in communications between Klarna and the customer. (max 1024 characters) |
| quantity_unit | optional | string(8) | Unit used to describe the quantity, e.g. kg, pcs... If defined has to be 1-8 characters |

| | | | |
|---|---|---|---|
| **product_identifiers** | optional | | List with product identifiers |
| brand | optional | string(255) | The product's brand name as generally recognized by consumers. If no brand is available for a product, do not supply any value |
| category_path | optional | string(255) | The product's category path as used in the merchant's webshop. Include the full and most detailed category and separate the segments with ' > ' |
| global_trade_item_number | optional | string(255) | The product's Global Trade Item Number (GTIN). Common types of GTIN are EAN, ISBN or UPC. Exclude dashes and spaces, where possible |
| manufacturer_part_number | optional | string(255) | The product's Manufacturer Part Number (MPN), which - together with the brand - uniquely identifies a product. Only submit MPNs assigned by a manufacturer and use the most specific MPN possible |
| **merchant_data** | optional | | List with merchant data |
| marketplace_seller_info | optional | string(255) | Information for merchant marketplace |

`required* = conditionally required`

Supported item types:

| Item Types |
|---|
| physical |
| discount |
| shipping_fee |
| sales_tax |
| digital |
| gift_card |
| store_credit |
| surcharge |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>invoice_refund</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <timestamp>2025-02-26T16:01:46Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>60</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>invoice_refund</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>420</code>
  <technical_message>can not do void on void reference</technical_message>
  <message>can not do void on void reference</message>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>60</amount>
  <currency>EUR</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |

| | | |
|---|---|---|
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

#### BITPAY REFUND

BitPay Refund is a custom refund method which will handle the asynchronous BitPay refund workflow. BitPay refunds can only be done on former transactions. Therefore, the reference id for the corresponding BitPay Sale transaction is mandatory.

BitPay Refund is an asynchronous transaction type.

When a BitPay Refund is requested, BitPay will send an email to the consumer with a request to provide the refund crypto address. This request will be valid for 3 days and will expire afterwards. When the crypto address is provided, the refund will be processed (processing usually takes 24 hours).

A Notification will be sent to the Merchant when the Bitpay refund is completed.

> Only full refunds are supported at the moment.

> BitPay Refunds can be voided only in the 24-hour processing period.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>bitpay_refund</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <reference_id>43672</reference_id>
   <amount>100</amount>
   <currency>USD</currency>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **bitpay_refund** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| reference_id | required | string(32) | The reference_id must be a BitPay Sale transaction |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>bitpay_refund</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <timestamp>2025-02-26T16:01:47Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
 </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

Error Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>bitpay_refund</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>410</code>
  <technical_message>no approved reference transaction found</technical_message>
  <message>no approved reference transaction found</message>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
</payment_response>
```


**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| code | integer | Error code according to Error code table |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |

#### PARTIAL REVERSAL

Partial reversal transactions are used in the preauthorization workflow to release a part of the total authorized amount.

> For more information, navigate to the Preauthorization Partial Reversal section in the preauthorization workflow.

---

## PayByLink

PayByLink is a frictionless payment via link. It provides merchants the ability to send a payment link to a customer via email or SMS (configurable), using Virtual Terminal.

#### WORKFLOW

The merchant can generate a PayByLink payment from the Virtual Terminal (if feature is enabled) by customizing an email template for sending payment link or configuring preferred medium (email/SMS) for sending payment link. When all the needed fields are filled in and the mandatory initial payload is provided, a payment request to the WPF is initiated. As a result of this request (if successful) the response would include redirect URL, which would be sent either via email or SMS to the customer (channel and needed email/phone number should be provided by the merchant when initiating the PayByLink payment). The customer can complete the payment by following the provided URL. It redirects to the merchant's web payment form (WPF) where the redirection workflows are the same as the ones described in the WPF section. The WPF reconcile API could also be used to check the status of any PayByLink initiated payment.

#### COMBINATION WITH PAY LATER FUNCTIONALITY

The PayByLink triggered payments could be easily combined with the 'Pay Later' functionality (available also for the WPF API). The PayByLink form provides the ability to choose whether the customer would have the option enabled to delay the payment and complete it later. It also gives the ability of enqueuing reminders based on pre-configured values. The reminders include the URL for payment completion as well.

#### REMINDERS CONFIGURATION

- Up to 3 reminders can be configured for each payment.
- The available channels for sending reminders are `email` and `sms`.
- The time for sending a reminder is set in number of minutes after payment creation.
- The time for sending of each reminder shouldn't be greater that the configured payment lifetime.

For configuration options or any other additional questions you can always contact Tech Support at tech-support@e-comprocessing.com.

---

## Alternative Payment Method External Events

#### INTRODUCTION

For some alternative payment method (APM) transactions additional events may occur resulting from various actions from the part of the consumer or the merchant. Examples: returned/reversed bank transfers, funds not received, additional bank transfers made using the same transaction reference number etc.

In Genesis these are called external events and are handled in the following manner:

1. A transaction note is created for the external event (visible under the ”Transaction Notes” section on the corresponding payment transaction page in the merchant console).
2. An API notification is sent to the merchant notification endpoint
3. An email notification is sent to the merchant admin email address
4. The original transaction status might be updated depending on the nature of the external event

#### LIST OF EXTERNAL EVENTS PER ALTERNATIVE PAYMENT METHOD

| APM | External event | Description | Status change |
|---|---|---|---|
| InstaDebit Payin | instadebit_payin_return | Payment has been returned to the consumer | voided |
| InstaDebit Payin | instadebit_payin_adjustment | Payment has been adjusted | none |
| InstaDebit Payout | instadebit_payout_return | Payment has been returned to the consumer | voided |
| InstaDebit Payout | instadebit_payout_adjustment | Payment has been adjusted | none |
| iDebit Payin | idebit_payin_return | Payment has been returned to the consumer | voided |
| iDebit Payin | idebit_payin_adjustment | Payment has been adjusted | none |
| iDebit Payout | idebit_payout_return | Payment has been returned to the consumer | voided |
| iDebit Payout | idebit_payout_adjustment | Payment has been adjusted | none |
| P24 | p24_external_refund | Payment has been rescinded by the consumer or was never received | refunded |

| | | | |
|---|---|---|---|
| Argencard | chargeback_external_event | Payment has been considered fraudulent by the card issuer and was reversed | chargebacked |
| Aura | chargeback_external_event | Payment has been considered fraudulent by the card issuer and was reversed | chargebacked |
| Cabal | chargeback_external_event | Payment has been considered fraudulent by the card issuer and was reversed | chargebacked |
| Cencosud | chargeback_external_event | Payment has been considered fraudulent by the card issuer and was reversed | chargebacked |
| Elo | chargeback_external_event | Payment has been considered fraudulent by the card issuer and was reversed | chargebacked |
| Naranja | chargeback_external_event | Payment has been considered fraudulent by the card issuer and was reversed | chargebacked |
| Nativa | chargeback_external_event | Payment has been considered fraudulent by the card issuer and was reversed | chargebacked |
| Tarjeta Shopping | chargeback_external_event | Payment has been considered fraudulent by the card issuer and was reversed | chargebacked |

**EMAIL NOTIFICATION**

An email is sent to the merchant admin's email address detailing the external event type and any other relevant details together with a link to the original payment transaction.

**API NOTIFICATION**

Merchants will receive API notifications every time Genesis obtains information about APM external events. Notifications are transmitted via HTTP POST (application/x-www-form-urlencoded) to the notification url endpoint provided in the XML request or to the Notification URL from the merchant account

An example notification:

```
&notification_type=apm_external_event
&signature=8216b1b4929c0a41ed44bf51726c10872a78f181b26d1427e15410da56803fc0a2f
&payment_transaction_unique_id=64216236bced683952325b6698b3954a
&category=citadel_payin_chargeback
&priority=info
&code=SDR
&info=Revoked+payment+due+to+instant+payment+funds+not+received
&message=A+Citadel+Payin+transaction+has+been+reversed%2Frevoked.+Payment+has+been+rescinded+by+the+consumer+or+funds+were+never+received+for+the+payment.
&payload=...
```

**Parameters:**

| Name | Type | Description |
|---|---|---|
| notification_type | string | constant value "apm external event" |
| signature | string | the signature of the notification, should be used to verify the the notification was sent by Genesis |
| payment_transaction_unique_id | string | unique id of the original transaction, generated by Genesis |
| category | string | type of the external event |
| priority | string | can be one of "info", "normal" or "urgent" |
| code | string | code for the external event from the APM provider system |
| info | string | a short description of the external event |
| message | string | full short description of the external event |
| payload | string | the raw response for the external event as received from the APM provider |

The signature is a security measure meant to ensure that the gateway is really the sender of the notification. It is generated by concatenating the unique id of the payment with your API password and generating a SHA-512 Hash (Hex) of the string:

`SHA-512 Hash Hex of [payment_transaction_unique_id][Your Merchant API password]`

**Notification signature examples**

| payment_transaction_unique_id | API password | signature |
|---|---|---|
| 26aa150ee68b1b2d6758a0e6c44fce4c | 50fd87e65eb415f42fb5af4c9cf497662e00b785 | c5219b3d385e74496b2b48a549 |
| 3f760162ef57a829011e5e2379b3fa17 | 50fd87e65eb415f42fb5af4c9cf497662e00b785 | 14519d0db2f7f8f407efccc9b09 |

```
<?xml version="1.0" encoding="UTF-8"?>
<notification_echo>
  <unique_id>3f760162ef57a829011e5e2379b3fa17</unique_id>
</notification_echo>
```

When receiving the notification, you are required to render an XML page containing the transaction's payment transaction unique id so that the gateway knows that you have accepted the notification. If the XML is not delivered, the notification is sent periodically as per the rules for notifications delivery.

---

# Advanced risk management with RiskParams

The risk params section in the payment transaction xml allows you to pass user specific values along with the payment transaction. These values may be used by advanced risk management features and checked against a blacklist.

RiskParams can be used in any user triggered payment transaction. User triggered transactions types are Authorize, Authorize3d, Sale, Sale3d, InitRecurringSale, InitRecurringSale3d, and AccountVerification.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
```

```
      <state>CA</state>
      <country>US</country>
    </billing_address>
    <risk_params>
      <ssn>987-65-4320</ssn>
      <mac_address>12-34-56-78-9A-BC</mac_address>
      <session_id>1DA53551-5C60-498C-9C18-8456BDBA74A9</session_id>
      <user_id>1002547</user_id>
      <user_level>vip</user_level>
      <email>test@example.com</email>
      <phone>+49301234567</phone>
      <remote_ip>245.253.2.12</remote_ip>
    </risk_params>
  </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **risk_params** | optional | | |
| ssn | optional | string(128) | Social Security number or equivalent value for non US customers. |
| mac_address | optional | string(128) | The customers mac address. |
| session_id | optional | string(128) | The customers session_id. |
| user_id | optional | string(128) | The customers user_id. |
| user_level | optional | string(128) | A value describing the customers trust level, may be used by the risk management for configurable differentiated limits. |
| email | optional | string(128) | The customers email. |
| phone | optional | string(128) | The customers phone. |
| remote_ip | optional | string(128) | The customers ip address. |
| serial_number | optional | string(128) | Custom serial number. |
| pan_tail | optional | string(128) | The last 4 digits of the card number. |
| bin | optional | string(128) | The first 6 digits of the card number. |
| first_name | optional | string(128) | Customer first name. |
| last_name | optional | string(128) | Customer last name. |
| country | optional | string(128) | The country of the customer. |
| pan | optional | string(128) | The PAN hash of the customer card number. |
| forwarded_ip | optional | string(128) | MaxMind specific risk param. |
| username | optional | string(128) | MaxMind specific risk param. |
| password | optional | string(128) | MaxMind specific risk param. |
| bin_name | optional | string(128) | MaxMind specific risk param. |
| bin_phone | optional | string(128) | MaxMind specific risk param. |

`required* = conditionally required`

> The risk management needs to be configured to use these values, passing the values alone will not trigger any risk management features.

To use these values for risk management please contact our Risk team.

---

## Credential On File (COF)

As the payment ecosystem has evolved, instances in which a transaction is initiated with a stored credential based on cardholder consent for future use have significantly increased. Growth in digital commerce, together with the emergence of new business models, has increased the number of transactions where a merchant or its agent, a payment facilitator (PF), or a staged digital wallet operator (SDWO) uses cardholder payment credentials (i.e., account details) that they previously stored for future purchases.

In Genesis, the COF indicator can be used for the following transaction types: Account Verification, Authorize, Authorize3D, Sale, Sale3D, InitRecurringSale, InitRecurringSale3D, Payout to mark a transaction as **initial customer initiated, subsequent customer initiated or as unscheduled merchant initiated (UCOF)**.

The UCOF transaction uses a previously stored credential for a fixed or variable amount and it does not occur on a scheduled or regularly occurring transaction date. With it, the cardholder has provided consent to the merchant to initiate one or more future transactions. An example of such a transaction is an account auto-top up.

**Supported options for Credential On File (COF) field:**

| COF | Description |
|---|---|
| initial_customer_initiated | Initial transaction used to store payment credentials for future customer initiated payments while processing. Required for external tokenization, and optional for gateway-based tokenization |
| subsequent_customer_initiated | Subsequent customer initiated transaction using previously stored payment credentials. Required for external tokenization, and optional for gateway-based tokenization |
| merchant_unscheduled | For UCOF transaction, the scheme transaction identifier of the initial transaction must be sent in the transaction request. For MasterCard or Maestro UCOF, the scheme settlement date in MMDD format (e.g. 0227) of the initial transaction must be sent in the transaction request. |

---

## Currency and Amount Handling

The gateway handles all types of processing currencies, with exponents ranging from 0 (e.g. JPY), 2 (e.g. CNY, USD, EUR, GBP), to 3 (e.g. KWD). Processing currencies are configured on terminal level.

Transaction amounts on the API level should be submitted in the minor currency unit for the given currency, e.g.:

**Amount currencies:**

| Name | Type | Description |
|---|---|---|
| USD | 100.33 | Should be submitted as 10033 in the amount API field (exponent 2) |
| EUR | 3 | Should be submitted as 300 in the amount API field (exponent 2) |
| JPY | 150 | Should be submitted as 150 in the amount API field (exponent 0) |
| KWD | 100.333 | Should be submitted as 100333 in the amount API field (exponent 3) |

**Amount limits:** Amount has to be provided within limit for listed transaction types and currencies:

| Transaction Type | Currency | minimum | maximum |
|---|---|---|---|
| Global limit | All | 0.01 | 1,000,000,000.00 |
| alipay | CNY | 0.01 | 50,000.00 |
| | EUR | 0.01 | 6,529.00 |
| davivienda | USD | 0.01 | 3,000.00 |
| banco de chile | USD | 0.01 | 3,000.00 |
| webpay | USD | 0.01 | 3,000.00 |
| pago facil | USD | 0.01 | 3,000.00 |
| rapi pago | USD | 0.01 | 3,000.00 |
| link | USD | 0.01 | 3,000.00 |
| santander | USD | 0.01 | 3,000.00 |
| aura | USD | 0.01 | 3,000.00 |
| cabal | USD | 0.01 | 3,000.00 |
| nativa | USD | 0.01 | 3,000.00 |
| naranja | USD | 0.01 | 3,000.00 |
| cencosud | USD | 0.01 | 3,000.00 |
| tarjeta shopping | USD | 0.01 | 3,000.00 |
| redpagos | USD | 0.01 | 3,000.00 |
| bcmc | EUR | 1,00 | 1,000,000.00 |
| elo | USD | 0.01 | 3,000.00 |
| oxxo | USD | 0.01 | 3,000.00 |
| bradesco | USD | 0.01 | 3,000.00 |
| cartao mercado livre | USD | 0.01 | 3,000.00 |
| efecty | USD | 0.01 | 3,000.00 |
| boleto | USD | 2.50 | 2,500.00 |
| itau | USD | 0.01 | 3,000.00 |
| multibanco | USD | 0.01 | 99,999.99 |
| banco do brasil | USD | 0.01 | 3,000.00 |
| argencard | USD | 0.01 | 3,000.00 |
| banco de occidente | USD | 0.01 | 3,000.00 |
| bancomer | USD | 0.01 | 3,000.00 |
| baloto | USD | 0.01 | 3,000.00 |
| eps | EUR | 1.00 | 1,000,000.00 |
| sofort | EUR | 1.00 | 5,000.00 |
| sdd sale | All | 0.10 | 24,999.99 |
| sct payout | All | 0.10 | 24,999.99 |
| sdd init recurring sale | All | 0.10 | 24,999.99 |
| sdd refund | All | 0.10 | 24,999.99 |
| neosurf | All | 0.01 | 9,999.99 |
| p24 | EUR | 0.01 | 15,000.00 |
| rpn payment | All | 0.10 | 100,000.00 |
| rpn payout | All | 100.00 | 1,000,000.00 |
| citadel payin | EUR | 0.01 | 10,000.00 |
| citadel payout | EUR | 0.01 | 10,000.00 |
| idebit payin | CAD | 0.01 | 1,500.00 |
| idebit payout | CAD | 0.01 | 1,500.00 |
| online banking | CNY* | 10.00 | 50,000.00 |
| | THB | 10.00 | 500,000.00 |
| | IDR | 10,000.00 | 50,000,000.00 |
| | MYR | 10.00 | 20,000.00 |
| bank payout | CNY | 60.00 | 49,000.00 |
| | THB | 350.00 | 175,000.00 |
| | IDR | 50,000.00 | 25,000,000.00 |
| | MYR | 50.00 | 20,000.00 |
| wechat | All | 10.00 | 3,000.00 |
| ezeecard payout | EUR | 10.00 | 800.00 |
| paysafecard | EUR | 0.01 | 1,000.00 |
| poli | AUD | 0.01 | 9,999.00 |

| | | | |
|---|---|---|---|
| insta debit payin | CAD | 0.01 | 1,500.00 |
| bitpay | USD | 1.00 | 950,000.00 |
| bitpay sale | USD | 1.00 | 950,000.00 |
| bitpay payout | USD | 1.00 | 950,000.00 |
| pse | USD | 0.01 | 3,000.00 |

`* = Depends on the setup`

Check the ISO 4217 standard for details on currencies and their exponents/minor currency units.

---

## Dynamic Descriptor

Dynamic descriptor functionality is available as part of the gateway. It is enabled on terminal level, so contact the IT Support team at tech-support@e-comprocessing.com if you wish to use it.

Currently, the transactions types that support dynamic descriptor parameters are Authorize, Authorize3d, Sale, Sale3d, InitRecurringSale, InitRecurringSale3d and Payout.

WPF payments also support dynamic descriptor.

The currently supported dynamic descriptor parameters are:

| Name | Type | Description |
|---|---|---|
| merchant_name | string(25) | Needed by merchants/PSPs to change the charge description. |
| merchant_city | string(13) | Contains the city of the merchant or the merchant phone number for CNP merchants. For master or Intl Maestro%, send the phone number in 'merchant_service_phone' field. |
| merchant_country | string(3) | Country code of the merchant country in ISO 3166 format. |
| merchant_state | string(3) | The value should be the merchant country subdivision code defined in ISO 3166-2. Invalid values submitted NOT as per the ISO might lead to data integrity issues. |
| merchant_zip_code | string(10) | Merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | string(48) | Merchant address. |
| merchant_url | string(60) | Merchant url. |
| merchant_phone | string(16) | Merchant phone. |
| merchant_service_city | string(13) | Merchant service city. |
| merchant_service_country | string(3) | Country code of the merchant service country in ISO 3166 format. |
| merchant_service_state | string(3) | The value should be the merchant service country subdivision code defined in ISO 3166-2. Invalid values submitted NOT as per the ISO might lead to data integrity issues. |
| merchant_service_zip_code | string(10) | Merchant service zip/postal code. |
| merchant_service_phone | string(16) | Merchant service phone. |
| sub_merchant_id | string(15) | Sub-merchant ID assigned by the Payment Facilitator. |
| merchant_geo_coordinates | string(20) | Merchant service geographic coordinates. Length of geographic coordinates is a range of 15-20 symbols. Latitude and Longitude separated by a comma. Example: `40.73061, -73.93524` |
| merchant_service_geo_coordinates | string(20) | Merchant service geographic coordinates. Length of geographic coordinates is a range of 15-20 symbols. Latitude and Longitude separated by a comma. Example: `40.73061, -73.93524` |

If the terminal is configured with dynamic descriptor, but the merchant does not send the dynamic descriptor parameter in question at all or sends whitespace, this defaults the given dynamic descriptor parameter to the static descriptor configured on MID level when submitted to the schemes.

If the value is less than max chars, it is right padded with whitespace before sent to the schemes. Note also that the dynamic descriptor params section with the properly formatted individual dynamic descriptor params is returned in the payment transaction response on the API if those params have been submitted in the transaction API request beforehand.

Regarding reference-based transactions, since they do not accept any dynamic descriptor params in the request, there is no dynamic descriptor params section in the response as well. Reference-based transactions reuse the dynamic descriptor params of their original transaction.

The sub-merchant ID should be provided by all merchants that are processing under Payment Facilitator. If the terminal does not support dynamic descriptor, a static value can be configured on a merchant level.

---

## Electronic Commerce Indicator

Electronic Commerce Indicator (ECI) is a value that is returned from the Directory Server (Visa, MasterCard, etc.) to indicate the authentication results of your customer’s credit card payment on 3D Secure.

Visa/JCB/Diners/American Express/Rupay

| ECI Code | Description |
|---|---|
| 05 | Both cardholder and card issuing bank are 3D enabled. 3D card authentication is successful |
| 06 | Either cardholder or card issuing bank is not 3D enrolled. 3D card authentication is unsuccessful, in sample situations as: 1. 3D cardholder not enrolled 2. Card issuing bank is not 3D Secure ready |
| 07 | Authentication is unsuccessful or not attempted. The credit card is either a non-3D card or card issuing bank does not handle it as a 3D transaction |

MasterCard/Maestro

| ECI Code | Description |
|---|---|
| 02 | Both cardholder and card issuing bank are 3D enabled. 3D card authentication is successful |
| 01 | Either cardholder or card issuing bank is not 3D enrolled. 3D card authentication is unsuccessful, in sample situations as: 1. 3D Cardholder not enrolled 2. Card issuing bank is not 3D Secure ready |
| 00 (or empty) | Authentication is unsuccessful or not attempted. The credit card is either a non-3D card or card issuing bank does not handle it as a 3D transaction |

---

## Scheme Response Codes

See below a list of Scheme response codes with the corresponding messages. Scheme response codes (scheme response code element) are different than the regular gateway codes (code element and response code element) - the scheme response code maps to the code returned by the schemes while the response code and code elements are acquirer or gateway internal code mapping and part of the API, as described in the Errors section. Transaction responses will return the relevant scheme response code (note that both scheme response code and authorization code are optional in the API responses and will be returned only if provided)

- Visa
- Mastercard and Maestro

**VISA**

| Code | Description |
|---|---|
| 00 | Approved or completed successfully |
| 01 | Refer to card issuer |
| 02 | Refer to card issuer, special condition |
| 03 | Invalid merchant |
| 04 | Pickup card (no fraud) |
| 05 | Do not honour |
| 06 | Error |
| 07 | Pick up card, special condition (fraud account) |
| 10 | Partial Approval |
| 11 | Approved (V.I.P) |
| 12 | Invalid transaction |
| 13 | Invalid amount or currency conversion field overflow |
| 14 | Invalid account number (no such number) |
| 15 | No such issuer |
| 19 | Re-enter transaction |
| 21 | No action taken |
| 25 | Unable to locate record in file |
| 28 | File temporarily not available for update or unquiry |
| 39 | No credit account |
| 41 | Lost card, pick up (fraud account) |
| 43 | Stolen card, pick up (fraud account) |
| 46 | Closed account |
| 51 | Not sufficient funds |
| 52 | No checking account |
| 53 | No savings account |
| 54 | Expired card or expiration date is missing |
| 55 | Incorrect PIN or PIN missing |
| 57 | Transaction not permitted to cardholder |
| 58 | Transaction not allowed at terminal |
| 59 | Suspected fraud |
| 61 | Exceeds approval amount limit |
| 62 | Restricted card (card invalid in this region or country) |
| 63 | Security violation (source is not correct issuer) |
| 64 | Transaction does not fulfil AML requirement |
| 65 | Exceeds withdrawal frequency limit |
| 70 | PIN data required |
| 74 | Different value than that used for PIN encryption errors |
| 75 | Allowable number of PIN entry tries exceeded |
| 76 | Unsolicited reversal |
| 78 | “Blocked, first used”—Transaction from new cardholder, and card not properly unblocked |
| 79 | Already reversed (by Switch) |
| 80 | No financial impact |
| 81 | Cryptographic error found in PIN |
| 82 | Negative CAM, dCVV, iCVV, or CVV results |
| 85 | No reason to decline a request for address verification, CVV2 verification, or a credit voucher or merchandise return |
| 86 | Cannot verify PIN; for example, no PVV |
| 89 | Ineligible to receive financial position information (GIV) |
| 91 | Issuer or switch inoperative and STIP not applicable or not available for this transaction; Time-out when no stand-in; POS Check Service: Destination unavailable; Credit Voucher and Merchandise Return Authorizations: V.I.P. sent the transaction to the issuer, but the issuer was unavailable. |
| 92 | Financial institution or intermediate network facility cannot be found for routing (receiving institution ID is invalid) |
| 93 | Transaction cannot be completed - violation of law |
| 96 | System malfunction |
| 1A | Additional customer authentication required. Soft decline. |
| 6P | Verification data failed |
| B1 | Surcharge amount not permitted on Visa cards or EBT food stamps (U.S. acquirers only) |
| B2 | Surcharge amount not supported by debit network issuer. |
| N0 | Force STIP |
| N3 | Cash service not available |
| N4 | Cash request exceeds issuer or approved limit |

| | |
|---|---|
| N5 | Ineligible for resubmission |
| N7 | Decline for CVV2 failure |
| N8 | Transaction amount exceeds preauthorized approval amount |
| P5 | Denied PIN unblock—PIN change or unblock request declined by issuer |
| P6 | Denied PIN change—requested PIN unsafe |
| Q1 | Card Authentication failed |
| R0 | Stop Payment Order |
| R1 | Revocation of authorisation order |
| R2 | Transaction does not qualify for Visa PIN |
| R3 | Revocation of all authorizations order |
| Z3 | Unable to go online; offline-declined |

**MASTERCARD AND MAESTRO**

| ISO Code | Description |
|---|---|
| 00 | Approved or completed successfully |
| 01 | Refer to card issuer |
| 03 | Invalid merchant |
| 04 | Capture card |
| 05 | Do not honour |
| 08 | Honour with ID |
| 10 | Partial Approval |
| 12 | Invalid transaction |
| 13 | Invalid amount |
| 14 | Invalid card number |
| 15 | Invalid issuer |
| 30 | Format error |
| 41 | Lost card |
| 43 | Stolen card |
| 51 | Insufficient funds/over credit limit |
| 54 | Expired card |
| 55 | Invalid PIN |
| 57 | Transaction not permitted to issuer/cardholder |
| 58 | Transaction not allowed at acquirer/terminal |
| 61 | Exceeds withdrawal amount limit |
| 62 | Restricted card |
| 63 | Security violation |
| 65 | Exceeds withdrawal count limit OR Identity Check Soft-Decline of EMV 3DS Authentication (merchant should Resubmit authentication with 3DS) |
| 70 | Contact Card Issuer |
| 71 | PIN Not Changed |
| 75 | Allowable number of PIN tries exceeded |
| 76 | Invalid/nonexistent “To Account” specified |
| 77 | Invalid/nonexistent “From Account” specified |
| 78 | Invalid/nonexistent account specified (general) |
| 79 | Life cycle (Mastercard use only) |
| 80 | System not available |
| 81 | Domestic Debit Transaction Not Allowed (Regional use only) |
| 82 | Policy (Mastercard use only) |
| 83 | Fraud/Security (Mastercard use only) |
| 84 | Invalid Authorization Life Cycle |
| 85 | Not declined |
| 86 | PIN Validation not possible |
| 87 | Purchase Amount Only, No Cash Back Allowed |
| 88 | Cryptographic failure |
| 89 | Unacceptable PIN—Transaction Declined—Retry |
| 90 | Cutoff is in progress |
| 91 | Authorization System or issuer system inoperative |
| 92 | Unable to route transaction |
| 94 | Duplication transaction detected |
| 96 | System error |
| 1Z | Authorization System or issuer system inoperative |

## Scheme Account Number Indicator Codes

The table below lists the possible Account Number Indicator (ANI) codes returned by MasterCard for scheme-tokenized transactions like Apple Pay, Google Pay, and Tokenized e-commerce via the Processing and the Web Payment Form APIs. The code is provided in the `scheme_pan_indicator` attribute and indicates the type of PAN associated with the token. The Account Number Indicator is returned together with the `scheme_pan_tail` attribute.

| Account Number Indicator | Description |
|---|---|
| C | MasterCard Digital Enablement Service Secure Element Token |
| E | Embossed Account Number Provided by Issuer |
| F | MasterCard Digital Enablement Service Card on File Token |
| H | MasterCard Digital Enablement Service Cloud-based Payments Token |
| L | Pay with Rewards Loyalty Program Operator [LPO] Card |
| M | Primary Account Number |
| P | Contactless Account Number |
| R | Pay with Rewards Card |
| V | Virtual Card Number |

## Manually Reviewed Transactions

Under certain conditions, transactions can be stopped for manual review by the Risk team. This happens when the appropriate risk rules have been enabled for the merchant in question. Feel free to discuss enabling of manual reviews for transactions with our Risk team. Transactions that are stopped for manual review are returned with status ‘pending review’ in the API response, together with a detailed message specifying a manual review of this transaction. Note that transactions will be manually reviewed by the Risk team in the next 24 hours. In the process, each transaction will be manually approved or manually declined, and at this point the merchant will receive a notification with the status of the transaction, see Notifications.

Special case for manual reviewing is when the transaction is 3D async, in this case the merchant receives one notification if the transaction is manually declined by the Risk team, and two notifications if the transaction is manually approved. On manual approval, the first notification is sent once the transaction is manually approved and is sent for enrollment check to the MPI provider - the notification will contain status pending asyc together with the redirect url where the consumer needs to be redirected to by the merchant. The second notification is the standard 3D notification, once a consumer has been redirected to the given redirect url, has entered his/her MPI password, and contains the final status of the transaction whether it is approved by the issuer, or declined for invalid 3D password, and so on

The format of the first notification for manual review and following approval is:

```
?transaction_id=82803B4C-70CC-43BD-8B21-FD0395285B40
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=sale3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=pending_async
&amount=500
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&eci=06
&redirect_url=http://example.com/redirect_url
```

The second notification is the same as the typical 3D notifications for final statuses after the consumer has performed the 3D workflow.

## Partial Approvals

Credit cards that do not have sufficient funds in their account for the full purchase amount may be provided with a partial approval response from the issuer. When a partial approval happens, there will be a flag confirming the partial approval in the response (partial approval set to ‘true’) and the amount field in the response will contain the actual partially approved amount instead of the requested one in the API request. The cardholder can then choose to use a supplemental payment method to pay the balance and complete the purchase, if so desired. Merchants that accept partial approvals should note that issuers may return a partial approval response on a pre-paid/debit card at any time, and issuers may also respond with a partial approval response amount that is equal to the requested amount.

Have in mind that full/partial refunds or captures need to reference the partially approved initial transaction, you will get a workflow error if trying to capture or refund more than the partially approved amount. So make sure you check for the partial approval flag in the API response for the relevant transaction types, and handle follow-up transaction amounts properly.

With 3-D secure transactions, if a partial approval happened, the partially approved amount will be returned in the async API notification to the merchant. This is because actual communication with the acquirers happen after 3D authentication by the cardholder, in the initial API response the merchant will get the requested amount with the status ‘pending async’ for the transaction.

Note that partial approval support is disabled by default, feel free to contact our Risk team to enable it.

## Preauthorizations

Preauthorizations are used to request Approval for an estimated transaction amount because the final transaction amount will only be known some time later. This type of message is typically sent for transactions such as car rentals, hotel rooms and petrol. The reason for the preauthorization is to authenticate the card and the cardholder and also to check funds availability.

They are similar to the final authorizations, but have longer authorize time-frame and allow amount to be extended *(restricted per card brand)*.

**Basic Workflow**

- Preauthorization
- Incremental authorize
- Capture
- Full reversal
- Partial reversal

**PREAUTHORIZATION**

Preauthorization transaction can be submitted via normal Authorize or Authorize3d transaction with additional request param **preauthorization**.

To enable this, please contact tech-support@e-comprocessing.com.

Supported Card brands

**VISA**

Visa Lodging Preauthorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize</transaction_type>
   <transaction_id>TrxID_8f126a73263b1996df3326e4133f4506</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>5000</amount>
   <currency>USD</currency>
```

```
  <preauthorization>true</preauthorization>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4200000000000000</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>TrxID_8f126a73263b1996df3326e4133f4506</transaction_id>
  <unique_id>591b0387a6c952722eff5e2f913a8c78</unique_id>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <cvv_result_code>M</cvv_result_code>
  <authorization_code>2104</authorization_code>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>5000</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

Reconcile Visa Preauthorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<reconcile>
  <unique_id>591b0387a6c952722eff5e2f913a8c78</unique_id>
</reconcile>'
```

Successful Preauthorization Reconciliation Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <authorization_code>2104</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>591b0387a6c952722eff5e2f913a8c78</unique_id>
  <transaction_id>TrxID_8f126a73263b1996df3326e4133f4506</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>5000</amount>
  <currency>USD</currency>
  <card_brand>visa</card_brand>
  <card_number>420000...0000</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <preauthorization>true</preauthorization>
  <preauthorization_expires_at>2025-03-29T16:01:47Z</preauthorization_expires_at>
  <preauthorization_total_amount>5000</preauthorization_total_amount>
  <capturable_amount>5750</capturable_amount>
  <captured_amount>0</captured_amount>
  <reversed_amount>0</reversed_amount>
  <reversible_amount>5000</reversible_amount>
</payment_response>
```

- MCC Restriction - **NO**
- Authorization timeframe - **7 days** *(depends on the MCC and merchant region)*
- Authorize timeframe extension - **not supported**
- Capture tolerance - *percent* or *amount*. Navigate to the Capture section to learn more.

| MCC | Segment | Authorization timeframe | Amount tolerance |
|---|---|---|---|
| 3501-3999, 7011 | Lodging | 31 days | 15% |
| 3351-3500, 7512 | Car Rental | 31 days | 15% |
| 4411 | Steamship and Cruise Lines | 31 days | 15% |
| 7513 | Truck Rentals | 7 days | 15% |
| 7033 | Trailer Parks and Campgrounds | 7 days | 15% |
| 7519 | Motor Home and Recreational Vehicle Rentals | 7 days | 15% |
| 5552 | Electric Vehicle Charging | 7 days | 15% |
| 7523 | Parking and Garages | 7 days | 15% |

| | | | |
|---|---|---|---|
| 7394 | Equipment, Tool, Furniture and Appliance Rental | 7 days | none |
| 7999 | Recreation Services | 7 days | none |
| 7996 | Amusement Parks, Carnivals, Circuses, Fortune Tellers | 7 days | none |
| 5599 | Miscellaneous Automotive, Aircraft, and Farm Equipment Dealers | 7 days | none |
| 4457 | Boat Rentals and Leasing | 7 days | none |
| 5571 | Motorcycle Shops and Dealers | 7 days | none |
| 4111 | Local and Suburban Commuter, Passenger Transportation, including Ferries | 7 days[1] | 25 USD[3] |
| 4112 | Passenger Railways | 7 days[1] | 25 USD[3] |
| 4131 | Bus Lines | 7 days[1] | 25 USD[3] |
| 5812 | Eating Places and Restaurants | end of approval day[2] | 20% |
| 5813 | Drinking Places, Bars, Taverns, Cocktail Lounges, Nightclubs, Discotheques | end of approval day[2] | 20% |
| 4121 | Taxicabs and Limousines (Card-Absent Environment only) | end of approval day[2] | 20% |

**7 days[1]** - 7 days *(3 days for US merchant region)*

**end of approval day[2]** - end of approval day *(in the acquirer's timezone)*

**25 USD[3]** - a capture with amount up to 25 USD can be requested without a need of additional incremental authorization. Just in case the authorized amount is less than 25 USD *(15 USD for merchants in the US region)*. The respective amount will be exchanged to the transaction currency in case the currency is different than USD.

**MASTERCARD**

Master Lodging Preauthorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize</transaction_type>
   <transaction_id>TrxID_b3db822c8fb8646ae8438b71d2792046</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>5000</amount>
   <currency>USD</currency>
   <preauthorization>true</preauthorization>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>5555555555554444</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
 </payment_transaction>'
```

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>authorize</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>TrxID_b3db822c8fb8646ae8438b71d2792046</transaction_id>
   <unique_id>e878779e3061a2334f5e94faf17033b9</unique_id>
   <avs_response_code>5I</avs_response_code>
   <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
   <cvv_result_code>M</cvv_result_code>
   <authorization_code>848554</authorization_code>
   <response_code>00</response_code>
   <timestamp>2025-02-25T16:01:47Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>5000</amount>
   <currency>USD</currency>
   <sent_to_acquirer>true</sent_to_acquirer>
 </payment_response>
```

Reconcile Master Preauthorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <unique_id>e878779e3061a2334f5e94faf17033b9</unique_id>
 </reconcile>'
```

Successful Preauthorization Reconciliation Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>authorize</transaction_type>
   <status>approved</status>
   <authorization_code>848554</authorization_code>
   <retrieval_reference_number>016813015184</retrieval_reference_number>
   <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
   <response_code>00</response_code>
   <unique_id>e878779e3061a2334f5e94faf17033b9</unique_id>
   <transaction_id>TrxID_b3db822c8fb8646ae8438b71d2792046</transaction_id>
   <mode>live</mode>
   <timestamp>2025-02-25T16:01:47Z</timestamp>
   <descriptor>Descriptor one</descriptor>
```

```
  <amount>5000</amount>
  <currency>USD</currency>
  <card_brand>master</card_brand>
  <card_number>555555...4444</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <preauthorization>true</preauthorization>
  <preauthorization_expires_at>2025-03-27T16:01:47Z</preauthorization_expires_at>
  <preauthorization_total_amount>5000</preauthorization_total_amount>
  <capturable_amount>5000</capturable_amount>
  <captured_amount>0</captured_amount>
  <reversed_amount>0</reversed_amount>
  <reversible_amount>5000</reversible_amount>
</payment_response>
```

- MCC Restriction - **NO**
- Authorize timeframe - **30 days**
- Authorize timeframe extension - **supported via Incremental authorize**
- Capture tolerance - **YES**, but only for the MCCs below. Navigate to the Capture section to learn more.

| MCC | Segment | Authorization timeframe | Amount tolerance |
|---|---|---|---|
| 5812 | Eating Places, Restaurants | 30 days | 20% |
| 5814 | Fast Food Restaurants | 30 days | 20% |

**MAESTRO**

Intl Maestro Lodging Preauthorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>authorize</transaction_type>
  <transaction_id>TrxID_2da010d8923c01e6e40f256f91d07467</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>5000</amount>
  <currency>USD</currency>
  <preauthorization>true</preauthorization>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>6759411100000008</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
</payment_transaction>'
```

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>TrxID_2da010d8923c01e6e40f256f91d07467</transaction_id>
  <unique_id>0f5bd1483b0edc0417e9d2a16573dc75</unique_id>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <cvv_result_code>M</cvv_result_code>
  <authorization_code>395925</authorization_code>
  <response_code>00</response_code>
  <timestamp>2025-02-24T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>5000</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

Reconcile Intl Maestro Preauthorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<reconcile>
  <unique_id>0f5bd1483b0edc0417e9d2a16573dc75</unique_id>
</reconcile>'
```

Successful Preauthorization Reconciliation Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <authorization_code>395925</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
```

```
  <response_code>00</response_code>
  <unique_id>0f5bd1483b0edc0417e9d2a16573dc75</unique_id>
  <transaction_id>TrxID_2da010d8923c01e6e40f256f91d07467</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-24T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>5000</amount>
  <currency>USD</currency>
  <card_brand>Intl Maestro</card_brand>
  <card_number>675941...0008</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <preauthorization>true</preauthorization>
  <preauthorization_expires_at>2025-03-03T16:01:47Z</preauthorization_expires_at>
  <preauthorization_total_amount>5000</preauthorization_total_amount>
  <capturable_amount>5000</capturable_amount>
  <captured_amount>0</captured_amount>
  <reversed_amount>0</reversed_amount>
  <reversible_amount>5000</reversible_amount>
</payment_response>
```

- MCC Restriction - **NO**
- Authorize timeframe - **7 days**
- Authorize timeframe extension - **supported via Incremental authorize**
- Capture tolerance - **YES**, but only for the MCCs below. Navigate to the Capture section to learn more.

| MCC | Segment | Authorization timeframe | Amount tolerance |
|---|---|---|---|
| 5812 | Eating Places, Restaurants | 7 days | 20% |
| 5814 | Fast Food Restaurants | 7 days | 20% |

---

Reconcile the preauthorization to retrieve the Preauthorization specifics:

- Preauthorization expiration
- Total preauthorized amount
- Capturable amount
- Captured amount

#### INCREMENTAL AUTHORIZE

Incremental authorizations are used in preauthorization workflow to:

- extend the preauthorization amount
- extend the preauthorization time-frame

> Incremental authorizations are non-3DS, because they only refer to the Preauthorization transaction. They cannot be voided / refunded etc, can only modify/extend the related preauthorization.

An incremental authorization transaction can be submitted in case:

- Preauthorization is approved and preauthorization time-frame is not expired
- Preauthorization has not been captured

> A Reconciliation could be performed to find out when a particular preauthorization is about to expire

Extend Preauthorization Timeframe & Amount

Mastercard Incremental Authorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>incremental_authorize</transaction_type>
   <transaction_id>TrxID_26f717977def85539822a138191df974</transaction_id>
   <usage>20469237 extend hotel rezervation</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>1000</amount>
   <reference_id>e878779e3061a2334f5e94faf17033b9</reference_id>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **incremental_authorize** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer >= 0 | Topup amount in minor currency unit, see Currency and Amount Handling for details |
| reference_id | required | string(32) | Unique id of the corresponding preauthorization transaction |

`required* = conditionally required`

> Incremental authorize with **zero** amount is allowed only for Mastercard and Maestro transactions. It will extend only the preauthorization timeframe, but not the preauthorized amount.

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>incremental_authorize</transaction_type>
   <status>approved</status>
   <mode>live</mode>
   <transaction_id>TrxID_26f717977def85539822a138191df974</transaction_id>
```

```
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <avs_response_code>5I</avs_response_code>
  <avs_response_text>Response provided by issuer processor; Address information not verified</avs_response_text>
  <authorization_code>485335</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>1000</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ’true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>incremental_authorize</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>TrxID_26f717977def85539822a138191df974</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>400</code>
  <technical_message>Preauthorization has been captured, no further incremental authorizations allowed</technical_message>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>1000</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| dynamic_descriptor_params | section | Optional, returned only if dynamic desc params are submitted on the API. Note here that the formatted dyn desc params are returned - as they would be submitted to the schemes for settlement. |

Example Xml For Extending The Time Frame Only

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
```

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>incremental_authorize</transaction_type>
  <transaction_id>TrxID_26f717977def85539822a138191df974</transaction_id>
  <usage>204692378 extend the preauthorization validity timeframe</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>0</amount>
  <reference_id>591b0387a6c952722eff5e2f913a8c78</reference_id>
</payment_transaction>'
```

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>incremental_authorize</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>TrxID_26f717977def85539822a138191df974</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>400</code>
  <technical_message>Incremental authorizations with no financial impact are currently not supported by card brand</technical_message>
  <message>Something went wrong, please contact support!</message>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>0</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

Reconcile Preauthorization By Unique Id Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<reconcile>
  <unique_id>e878779e3061a2334f5e94faf17033b9</unique_id>
</reconcile>'
```

Preauthorization time-frame will be extended for Mastercard & Maestro transactions, but not for VISA

Successful Preauthorization Reconciliation Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <authorization_code>485335</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>e878779e3061a2334f5e94faf17033b9</unique_id>
  <transaction_id>TrxID_26f717977def85539822a138191df974</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>5000</amount>
  <currency>USD</currency>
  <card_brand>master</card_brand>
  <card_number>555555...4444</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <preauthorization>true</preauthorization>
  <preauthorization_expires_at>2025-04-26T16:01:47Z</preauthorization_expires_at>
  <preauthorization_total_amount>6000</preauthorization_total_amount>
  <capturable_amount>6000</capturable_amount>
  <captured_amount>0</captured_amount>
</payment_response>
```

**CAPTURE**

Preauthorizations can be captured using the standard Capture transaction with the correct `reference_id` of the Preauthorization.

A preauthorization can be captured only once. It can be a full & partial capture, but not multiple partial capture. In case of partial capture, the amount left needs to be reversed / returned first to the consumer using Partial Reversal transaction.

The capturable amount can be greater than the total authorized amount (only for VISA), check Preauthorization section for more info. The amount tolerance is supported only for VISA transactions, can be defined per percent or amount.

An additional incremental authorization will be necessary when the requested capture amount is greater than:

- the total authorized amount including the calculated amount tolerance (% of the total authorized amount)
- the predefined amount tolerance in value exchanged to the appropriate currency (check the amount tolerance mer MCC in the Preauthorization section)

Multiple partial captures are not allowed for both supported card brands

Reconcile Visa Preauthorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
```

```
<?xml version="1.0" encoding="UTF-8"?>
<reconcile>
  <unique_id>591b0387a6c952722eff5e2f913a8c78</unique_id>
</reconcile>'
```

Successful Preauthorization Reconciliation Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <authorization_code>005645</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ8RKA1EBL470G9XYG90R5D3E</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>591b0387a6c952722eff5e2f913a8c78</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>3000</amount>
  <currency>USD</currency>
  <card_brand>visa</card_brand>
  <card_number>420000...0000</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <preauthorization>true</preauthorization>
  <preauthorization_expires_at>2025-03-29T16:01:47Z</preauthorization_expires_at>
  <preauthorization_total_amount>5000</preauthorization_total_amount>
  <capturable_amount>5750</capturable_amount>
  <captured_amount>0</captured_amount>
  <reversed_amount>0</reversed_amount>
  <reversible_amount>5000</reversible_amount>
</payment_response>
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>capture</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>Check out from the Hotel</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <reference_id>43672</reference_id>
  <amount>5500</amount>
  <currency>USD</currency>
</payment_transaction>'
```

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>capture</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <authorization_code>345678</authorization_code>
  <response_code>00</response_code>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>5500</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
</payment_response>
```

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>capture</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>420</code>
  <technical_message>partial reversal is required first for the rest amount</technical_message>
  <message>Transaction declined.</message>
  <timestamp>2025-02-26T16:01:48Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>4000</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

Reconcile Visa Preauthorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<reconcile>
  <unique_id>591b0387a6c952722eff5e2f913a8c78</unique_id>
</reconcile>'
```

Successful Preauthorization Reconciliation Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <authorization_code>005645</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>591b0387a6c952722eff5e2f913a8c78</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>3000</amount>
  <currency>USD</currency>
  <card_brand>visa</card_brand>
  <card_number>420000...0000</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <preauthorization>true</preauthorization>
  <preauthorization_expires_at>2025-03-29T16:01:47Z</preauthorization_expires_at>
  <preauthorization_total_amount>5000</preauthorization_total_amount>
  <capturable_amount>0</capturable_amount>
  <captured_amount>5500</captured_amount>
  <reversed_amount>0</reversed_amount>
  <reversible_amount>0</reversible_amount>
</payment_response>
```

**FULL REVERSAL**

Full reversal of a preauthorization can be submitted using the standard Void transaction.

If a preauthorization has not been captured/cleared, the merchant must ensure to submit a full reversal not later than 24 hours after the preauthorization has expired, otherwise a full reversal will be automatically performed by Genesis.

A preauthorization can be fully reversed via the standard Void transaction only if it has not been captured or partially reversed via Partial Reversal transaction.

Reconcile request can be used to determine when the preauthorization is about to expire.

Reconcile Master Preauthorization Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <unique_id>e878779e3061a2334f5e94faf17033b9</unique_id>
 </reconcile>'
```

Successful Preauthorization Reconciliation Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <authorization_code>005645</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>e878779e3061a2334f5e94faf17033b9</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-25T16:01:47Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>3000</amount>
  <currency>USD</currency>
  <card_brand>master</card_brand>
  <card_number>555555...4444</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <preauthorization>true</preauthorization>
  <preauthorization_expires_at>2025-03-27T16:01:47Z</preauthorization_expires_at>
  <preauthorization_total_amount>5000</preauthorization_total_amount>
  <capturable_amount>5000</capturable_amount>
  <captured_amount>0</captured_amount>
  <reversed_amount>0</reversed_amount>
  <reversible_amount>5000</reversible_amount>
</payment_response>
```

**PARTIAL REVERSAL**

Partial reversal transactions are used in the preauthorization workflow to release a part of the total authorized amount. A transaction of this type should refer to the preauthorization directly.

A partial reversal cannot be performed for the full authorized amount. If you would like to reverse the entire amount, you would need to use a Void transaction.

The partial reversals can be submitted no later than 24 hours after the preauthorization is about to expire.

Reconcile could be performed to retrieve the reversible amount of a preauthorization. If a VISA Preauthorization has already been captured with a higher amount than the total preauthorized (benefiting from VISA amount tolerance), the reversible amount will be **0**. Otherwise, **reversible amount** = **preauthorization total amount** - **captured amount** - **reversed amount**.

Extend Preauthorization Timeframe & Amount

Partial Reversal Request Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>partial_reversal</transaction_type>
   <transaction_id>TrxID_df4bbf054f8d1d07928b2d55ec216c07</transaction_id>
   <usage>40208 hotel reservation changed</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>500</amount>
   <reference_id>e8787779e3061a2334f5e94faf17033b9</reference_id>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_type | required | string(255) | The transaction type: **partial_reversal** |
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | required* | string(255) | Description of the transaction for later use. |
| remote_ip | required* | IPv4 or IPv6 address | IPv4 or IPv6 address of customer |
| amount | required | integer > 0 | The amount to be reversed in minor currency unit, see Currency and Amount Handling for details. |
| reference_id | required | string(32) | Unique id of the corresponding preauthorization transaction |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>partial_reversal</transaction_type>
   <status>approved</status>
   <authorization_code>629324</authorization_code>
   <retrieval_reference_number>016813015184</retrieval_reference_number>
   <response_code>00</response_code>
   <transaction_id>TrxID_df4bbf054f8d1d07928b2d55ec216c07</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <timestamp>2025-02-26T16:01:48Z</timestamp>
   <mode>live</mode>
   <descriptor>Descriptor one</descriptor>
   <amount>500</amount>
   <currency>USD</currency>
   <sent_to_acquirer>true</sent_to_acquirer>
 </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| gaming | 'true' | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995, contact tech support for more details. |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ’true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |

Error Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>partial_reversal</transaction_type>
   <status>error</status>
   <mode>live</mode>
   <transaction_id>TrxID_df4bbf054f8d1d07928b2d55ec216c07</transaction_id>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <code>410</code>
```

```
    <technical_message>No approved preauthorization reference transaction found</technical_message>
    <message>Something went wrong, please contact support!</message>
    <timestamp>2025-02-26T16:01:48Z</timestamp>
    <descriptor>Descriptor one</descriptor>
    <amount>500</amount>
    <currency>USD</currency>
    <sent_to_acquirer>false</sent_to_acquirer>
  </payment_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| sent_to_acquirer | string(255) | "true" or "false" |

## Required vs Optional API params

There are some API params which can be configured as either optional or required on the terminal level. Contact the IT Support team at tech-support@e-comprocessing.com if you wish to disable or enable some API params based on your business model and integration type. Note also that if all the billing address params are configured as optional, then the whole billing address XML tag can be dismissed and not sent in the API requests.

| Name | Type | Description |
|---|---|---|
| customer_email | email address | By default, customer email is a required API param |
| customer_phone | string(32) | By default, customer phone is an optional API param |
| remote_ip | string(40) | By default, customer IP is a required API param |
| first_name | string(255) | By default, customer first name is a required API param |
| last_name | string(255) | By default, customer last name is a required API param |
| address1 | string(255) | By default, primary address is a required API param. Cannot be configured as optional for account verification transaction type |
| address2 | string(255) | By default, secondary address is an optional API param |
| city | string(255) | By default, city is a required API param. Cannot be config- ured as optional for account verification transaction type |
| zip_code | string | By default, zip code is a required API param. Cannot be con- figured as optional for account verification transaction type |
| state | string(2) | By default, state is an optional API param (unless country is US or CA) |
| country | string(2) | By default, country code is a required API param |
| usage | string(255) | By default, usage will be configured as a required API param for SDD and P24, optional for the rest |

## Transaction States

Transactions will have one of the following states. These will be returned by transaction responses, reconcile responses and will be shown in the browser interface

| Status | Description |
|---|---|
| approved | Transaction was approved by the schemes and is successful. |
| declined | Transaction was declined by the schemes or risk management. |
| pending_async | An asynchronous transaction (3-D secure payment) has been initiated and is waiting for user input. Updates of this state will be sent to the notification url specified in request. |
| pending_hold | An asynchronous transaction has been finalized by user but is waiting final update from provider. |
| pending | The outcome of the transaction could not be determined, e.g. at a timeout situation. Transaction state will eventually change, so make a reconcile after a certain time frame. |
| error | An error has occurred while negotiating with the schemes. |
| refunded | Once an approved transaction is refunded the state changes to refunded. |
| chargebacked | Once an approved transaction is chargebacked - the state changes to changebacked. Chargeback is the state of rejecting an accepted transaction (with funds transferred) by the cardholder or the issuer |
| voided | Transaction was authorized, but later the merchant canceled it. |
| chargeback_reversed | Once a chargebacked transaction is charged, the state changes to chargeback reversed. Chargeback has been canceled. |
| represented | Once a chargebacked transaction is charged, the state changes to represented. Chargeback has been canceled. |
| representment_reversed | Once a represented transaction is reversed, the state changes to representment reversed. |
| second_chargebacked | Once a chargeback_reversed transaction is chargebacked the state changes to second chargebacked. |
| pending_review | Transaction on hold, a manual review will be done |
| partially_reversed | Relevant only for preauthorization transactions. Once partial amount (less than the full preauthorized amount) is returned to the customer, the transaction becomes partially reversed. |

# Supported Card Brands

Card Brands are specific per acquirer. If you want to use a specific card brand you can contact tech-support@e-comprocessing.com.

| Supported card brands |
|---|
| Visa |
| Master |
| Intl Maestro |
| Discover |
| Diners |
| AMEX |
| JCB |
| RuPay |
| Elo |
| Aura |
| Hipercard |

---

# Document ID Parameter

Document ID is consumer personal identification. It is different for every country and is described more specifically in the table below. Document ID is required for some of the acquirers, please contact the IT Support team at tech-support@e-comprocessing.com for more information.

| Country | Name | Type | Description |
|---|---|---|---|
| Argentina | document_id | string(255) | Consumer's Argentinian Identification Number(DNI or CUIT). Must be string between 7 to 9, or 11 digits. |
| Brazil | document_id | string(255) | Consumer's Brazilian Identification Number(CPF or CNPJ). Must be string between 11 and 14 digits and to have full cpf validation. Example: 76484475687 |
| Chile | document_id | string(255) | Consumer's Chilean Identification Number(CI/RUT). Must be string between 8 to 9 digits. |
| Colombia | document_id | string(255) | Consumer's Colombian Identification Number(CC). Must be string between 6 to 10 digits. |
| India | document_id | string(255) | Consumer's Indian PAN. Must be string with 10 alphanumeric letters. 5 letters, followed by 4 numbers, followed by 1 letter or number. Example: ABCDE1234F |
| Mexico | document_id | string(255) | Consumer's Mexican Identification Number(CURP). Must be string between 10 and 18 digits. |
| Paraguay | document_id | string(255) | Consumer's Paraguayan Identification Number(CI). Must be string between 5 and 20 digits. |
| Peru | document_id | string(255) | Consumer's Peruvian Identification Number(DNI). Must be string between 8 and 9 digits. |
| Turkey | document_id | string(255) | Consumer's Turkish Identification Number(T.C. Kimlik No.). Must be string between 5 and 20 digits. |
| Uruguay | document_id | string(255) | Consumer's Uruguayan Identification Number(CI). Must be string between 6 and 8 digits. |

---

# Business Attributes

Business attributes are groups of additional risk attributes which are in close relation with the merchant business category. Some/All of them can be required at our risk team's discretion and will be used for internal reporting only. These business attributes can be submitted with a standard card transaction on Processing API and WPF processing.

**Business categories:**

| Segment | MCC |
|---|---|
| **Airlines Air Carriers** | |
| Airlines, Air Carriers | 4511 |
| Airlines | 3000 - 3302 |
| **Event Management** | |
| Consulting, Public Relations | 7392 |
| Miscellaneous General Services | 7299 |
| Theatrical Ticket Agencies | 7922 |
| Direct Marketing - Other | 5969 |
| **Furniture** | |
| Furniture, Home Furnishings, and Equipment Stores, Except Appliances | 5712 |
| Office and Commercial Furniture | 5021 |
| **Hotels and Real estate Rentals** | |
| Hotels/Motels/Inns/Resorts | 3501 - 3790 |
| Real Estate Agents and Managers - Rentals | 6513 |
| Lodging – Hotels, Motels, Resorts, Central Reservation Services (not elsewhere classified) | 7011 |
| Timeshares | 7012 |
| **Car, plane and Boat rentals** | |
| Car Rental | 3351 - 3441 |
| Taxicabs and Limousines | 4121 |
| Bus Lines, Including Charters, Tour Buses | 4131 |
| Boat Rentals and Leases | 4457 |
| Transportation Services, (Not elsewhere classified) | 4789 |

| | |
|---|---|
| Car Rental Companies | 7512 |
| Truck and Utility Trailer Rentals | 7513 |
| Motor Home and Recreational Vehicle Rentals | 7519 |
| **Cruise Lines** | |
| Cruise Lines | 4411 |
| **Travel Agencies** | |
| Travel Agencies | 4722 |
| Package Tour Operators (For use in Germany only) | 4723 |
| Direct Marketing - Travel-related Arrangement Services | 5962 |

Specific attributes for each business category can be found in the following table:

| Attribute | Type | Description |
|---|---|---|
| **Airlines Air Carriers** | | |
| flight_arrival_date | string | The date when the flight arrives in format dd-mm-yyyy |
| flight_departure_date | string | The date when the flight departs in format dd-mm-yyyy |
| airline_code | string | The code of Airline |
| airline_flight_number | string | The flight number |
| flight_ticket_number | string | The number of the flight ticket |
| flight_origin_city | string | The origin city of the flight |
| flight_destination_city | string | The destination city of the flight |
| airline_tour_operator_name | string | The name of tour operator |
| payment_type | string | The type of payment - can be either `deposit` or `balance` |
| **Event Management** | | |
| event_start_date | string | The date when event starts in format dd-mm-yyyy |
| event_end_date | string | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | string | |
| event_id | string | |
| payment_type | string | The type of payment - can be either `deposit` or `balance` |
| **Furniture** | | |
| date_of_order | string | The date when order was placed in format dd-mm-yyyy |
| delivery_date | string | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | string | |
| payment_type | string | The type of payment - can be either `deposit` or `balance` |
| **Hotels and Real estate rentals** | | |
| check_in_date | string | The data when the customer check-in in format dd-mm-yyyy |
| check_out_date | string | The data when the customer check-out in format dd-mm-yyyy |
| travel_agency_name | string | |
| payment_type | string | The type of payment - can be either `deposit` or `balance` |
| **Car, Plane and Boat Rentals** | | |
| vehicle_pick_up_date | string | The date when customer takes the vehicle in format dd-mm-yyyy |
| vehicle_return_date | string | The date when the customer returns the vehicle back in format dd-mm-yyyy |
| supplier_name | string | The name of supplier/contractor |
| payment_type | string | The type of payment - can be either `deposit` or `balance` |
| **Cruise Lines** | | |
| cruise_start_date | string | The date when cruise begins in format dd-mm-yyyy |
| cruise_end_date | string | The date when cruise ends in format dd-mm-yyyy |
| payment_type | string | The type of payment - can be either `deposit` or `balance` |
| **Travel Agencies** | | |
| arrival_date | string | The date of arrival in format dd-mm-yyyy |
| departure_date | string | The date of departure in format dd-mm-yyyy |
| carrier_code | string | The code of the carrier |
| flight_number | string | The number of the flight |
| ticket_number | string | The number of the ticket |
| origin_city | string | The origin city |
| destination_city | string | The destination city |
| travel_agency | string | The name of the travel agency |
| contractor_name | string | The name of the contractor |
| atol_certificate | string | ATOL certificate number |
| pick_up_date | string | Pick-up date in format dd-mm-yyyy |
| return_date | string | Return date in format dd-mm-yyyy |

| | | |
|---|---|---|
| payment_type | string | The type of payment - can be either `deposit` or `balance` |

**Transaction types with business attributes:**

- authorize
- authorize3d
- capture
- sale
- sale3d
- init_recurring_sale
- init_recurring_sale3d
- recurring_sale
- trustly_sale

---

# Recurring Advice

The recurring advice is an additional response code returned from the schemes. Specifies if the transaction can be retried in case of failure. Available codes:

| Recurring Advice Code | Recurring Advice Text |
|---|---|
| 01 | New Account Information available |
| 02 | Try again later |
| 03 | Do not try again |
| 04 | Token requirements not fulfilled for this token type |
| 21 | Recurring Payment Cancellation Service (the new fee applies with this one) |
| 22 | Merchant does not qualify for product code |
| 24 | Retry after 1 hour |
| 25 | Retry after 24 hours |
| 26 | Retry after 2 days |
| 27 | Retry after 4 days |
| 28 | Retry after 6 days |
| 29 | Retry after 8 days |
| 30 | Retry after 10 days |
| 31 | Retry later (max 15 attempts in 30 day period) |
| 40 | Consumer non-reloadable prepaid card |
| 41 | Consumer single-use virtual card number |
| 42 | Sanctions Scoring Service: Score Exceeds Applicable Threshold Value |
| 43 | Consumer multi-use Virtual Card Number (VCN) |

---

# Tokenized e-commerce

Visa Synchronous 3 D Sv2 Scheme Tokenized Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4012000000060085</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <scheme_tokenized>true</scheme_tokenized>
   <business_attributes>
     <event_start_date>27-03-2025</event_start_date>
     <event_end_date>08-04-2025</event_end_date>
     <event_organizer_id>20192375</event_organizer_id>
     <event_id>1912</event_id>
   </business_attributes>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <mpi_params>
     <eci>05</eci>
     <cavv>Y9Ri418AQCrkPp16sR+nMAACAAA=</cavv>
     <protocol_version>2</protocol_version>
     <protocol_sub_version>2</protocol_sub_version>
   </mpi_params>
 </payment_transaction>'
```

Master Synchronous 3 D Sv2 Scheme Tokenized Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
```

```
<payment_transaction>
  <transaction_type>authorize3d</transaction_type>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>5555555555559997</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <business_attributes>
    <event_start_date>27-03-2025</event_start_date>
    <event_end_date>08-04-2025</event_end_date>
    <event_organizer_id>20192375</event_organizer_id>
    <event_id>1912</event_id>
  </business_attributes>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <scheme_tokenized>true</scheme_tokenized>
  <mpi_params>
    <eci>02</eci>
    <cavv>AM1wRssmx0j1ABoxs1jIAoABFA==</cavv>
    <protocol_version>2</protocol_version>
    <protocol_sub_version>2</protocol_sub_version>
  </mpi_params>
</payment_transaction>'
```

Successful Synchronous Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize3d</transaction_type>
  <status>approved</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <authorization_code>345678</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <response_code>00</response_code>
  <timestamp>2025-02-26T16:01:48Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <threeds>
    <eci>05</eci>
  </threeds>
</payment_response>
```

E-commerce tokenization is a new way in the payment processing services that provides enhanced safety and convenience for all participants in the process. The technology is based on the replacement of sensitive payment data as PAN to DPAN (or token) and is known as Visa tokenization for Visa and Digital Secure Remote Payments (DSRP) for Mastercard.

> The tokenization is supported by processing API with the following transaction types: Authorize, Authorize3D, Sale, Sale3D, InitRecurringSale, InitRecurringSale3D

In the case of using scheme tokenization with DPAN instead of FPAN, you need to add a special param `scheme_tokenized` with a value of `true` in the transaction API request. There is also a clarification regarding the `mpi_params` section (for 3D transactions only):

- The `directory_server_id` param is not required
- The cryptogram is placed in the CAVV attribute inside the `mpi_params` and does not match any of the leading indicators for MasterCard Identity check.

On the right there are examples for Visa and Mastercard.

Scheme tokenized transactions are enabled on purpose, please contact Tech Support for more details.

---

# Customer Identification Parameters

Customer Identification Parameters give additonal information to the acquirer about the customer of the payment.

> Required for Visa OCT (Credit , Payout) transactions destined for Brazil or Qatar.

**OWNER**

Specifies if the document ID belongs to the sender or the receiver of the OCT.

| Valid values |
|---|
| sender |
| receiver |

**TYPE**

Specifies the type of the document ID.

| Valid values |
|---|
| birth_date |
| unspecified |
| national |
| passport_number |
| driver_license |
| tax |
| company_registration_number |

| |
|---|
| proxy |
| social_security_number |
| alien_registration_number |
| law_enforcement |
| military |
| travel |
| email |
| phone_number |

**SUBTYPE**

Specifies if the document ID is registered for business or individual usage.

| Valid values |
|---|
| business |
| individual |

**DOCUMENT ID**

The document ID of the customer. See Document ID Parameter for more details. In case the `document_id` has a `birth_date` type, the required format is `yyyy-mm-dd` and must contain the valid birth date of the customer.

**ISSUING COUNTRY**

The country that has issued the document ID and has to be a country code in ISO 3166.

---

## Scheme Maximum Settlement Date

The Scheme Maximum Settlement Date is a date that the schemes authorities compute and return with authorization response messages indicating the final date an authorization could be captured. This date defines the authorization hold period. The feature can be used with all authorization and full financial messages for purchase and purchase return transactions. Currently, this field is only supported for VISA transactions. When merchants submit authorizations with Visa, Visa will compute and provide a final date by when an authorization must be captured. If the Scheme Maximum Settlement Date has not passed, the capture will be approved, otherwise the capture will be declined with an error. The card scheme will provide the Maximum Settlement Date for sale transactions as well. Furthermore, the system will return the “scheme_max_settlement_date” parameter for purchase transactions.

---

# Reconcile

---

Reconcile can be used to retrieve data about a transaction. This can be useful if you want to retrieve information about a transaction whose status is timeout, which returned an error or has changed eg. has beed chargebacked.

Reconcile request are handled exactly like transaction requests via XML.

---

## Single Transaction

The URL for single transaction reconciling is similar to the processing url:

`https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN`

Reconcile By Unique Id Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
 </reconcile>'
```

Reconcile By Arn Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <arn>74537605248535042582882</arn>
 </reconcile>'
```

Reconcile By Transaction Id Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <transaction_id>merchant-transaction-id-here</transaction_id>
 </reconcile>'
```

**XML Request to reconcile:**

Note that reconcile can be done via either unique_id, ARN or transaction_id

**XML Response:**

Response is a standard payment response like it would be returned by any transaction. It can have either state as shown in the states section.

Successful Sale Transaction Reconciliation Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale</transaction_type>
  <status>approved</status>
  <authorization_code>005645</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:48Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>9000</amount>
  <currency>USD</currency>
  <card_brand>visa</card_brand>
  <card_number>420000...0000</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```


Card brand and card number will be available in response only for card transaction types.

The reference transaction unique id is also returned when a reference-based transaction has been queried via the Reconcile API.

Successful Refund Transaction Reconciliation Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>refund</transaction_type>
  <status>approved</status>
  <authorization_code>005645</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>5de39380bf7ac7e1fc31cbd7805dc0ec</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:48Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>9000</amount>
  <currency>USD</currency>
  <card_brand>visa</card_brand>
  <card_number>420000...0000</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <reference_transaction_unique_id>44177a21403427eb96664a6d7e5d5d48</reference_transaction_unique_id>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```


Successful Sale3d 3 D Sv2 Transaction Reconciliation Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>approved</status>
  <authorization_code>005645</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:48Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>9000</amount>
  <currency>USD</currency>
  <card_brand>visa</card_brand>
  <card_number>420000...0000</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <threeds>
    <authentication_flow>frictionless</authentication_flow>
    <threeds_method>
      <status>completed</status>
    </threeds_method>
    <protocol>
      <target_version>2</target_version>
      <concrete_version>2</concrete_version>
      <sub_version>2</sub_version>
```

```
    </protocol>
    <eci>05</eci>
  </threeds>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```


Captured flag and possible remaining capturable amount would be returned when an authorization transaction has been queried via the Reconcile API.

Successful Authorize Transaction Reconciliation Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <authorization_code>005645</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:48Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>500</amount>
  <currency>USD</currency>
  <card_brand>visa</card_brand>
  <card_number>420000...0000</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <captured>false</captured>
  <capturable_amount>500</capturable_amount>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```


For some transaction types current funds status will also be available in the reconcile response. Note that refunds on those transactions will have funds status on the following day of the transaction. For more information please contact tech-support.

Successful Transaction Reconciliation Response with funds status attribute


```
<?xml version='1.0' encoding='UTF-8'?>
  <payment_response>
    <transaction_type>aura</transaction_type>
    <status>approved</status>
    <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <code>940</code>
    <mode>live</mode>
    <technical_message>Transaction successful!</technical_message>
    <message>Transaction successful!</message>
    <timestamp>2020-11-03T07:25:10Z</timestamp>
    <descriptor>Descriptor one</descriptor>
    <amount>50</amount>
    <currency>USD</currency>
    <funds_status>SUCCEEDED</funds_status>
    <account_holder>Name+Surname</account_holder>
  </payment_response>
```


Declined Transaction Reconciliation Response


```
<?xml version='1.0' encoding='UTF-8'?>
  <payment_response>
    <transaction_type>sale3d</transaction_type>
    <status>declined</status>
    <exemption>low_risk</exemption>
    <sca_exemption_result>23</sca_exemption_result>
    <cvv_result_code>N</cvv_result_code>
    <retrieval_reference_number>433524286054</retrieval_reference_number>
    <scheme_response_code>N7</scheme_response_code>
    <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
    <transaction_id>119643250547501c79d8295</transaction_id>
    <response_code>N7</response_code>
    <code>340</code>
    <technical_message>CVV2 Failure</technical_message>
    <message>Please check input data for errors!</message>
    <mode>live</mode>
    <timestamp>2020-11-03T07:25:10Z</timestamp>
    <descriptor>eMPPay.com  *eMPPay</descriptor>
    <amount>100</amount>
    <currency>EUR</currency>
    <threeds>
      <eci>06</eci>
    </threeds>
    <sent_to_acquirer>true</sent_to_acquirer>
    <card_brand>visa</card_brand>
    <card_number>479687...8062</card_number>
    <card_type>DEBIT</card_type>
    <card_subtype>BUSINESS</card_subtype>
    <card_issuing_bank>DSK BANK PLC</card_issuing_bank>
    <card_issuing_country>bulgaria</card_issuing_country>
    <card_holder>John Doe</card_holder>
    <expiration_year>2033</expiration_year>
    <expiration_month>8</expiration_month>
    <scheme_transaction_identifier>119643250547501c79d8295</scheme_transaction_identifier>
  </payment_response>
```


**PREAUTHORIZATION**

Custom response data will be returned when a Preauthorization transaction has been queried via the Reconcile API.

| Name | Type | Description |
|---|---|---|
| preauthorization | "true" | Preauthorization flag |
| preauthorization_expires_at | string | Preauthorization expiration date time in ISO 8601 Combined date and time, e.g. 2007-08-30T17:46:11Z |
| preauthorization_total_amount | integer | Total preauthorization amount (initial + topup amount) |
| capturable_amount | integer | The total amount that can be captured |
| captured_amount | integer | The total captured amount |
| reversed_amount | integer | The total reversed amount |
| reversible_amount | integer | The total reversible amount |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |

The total capturable amount will be decreased in case there is/are already submitted partial reversal(s).

Successful Preauthorization Reconciliation Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize</transaction_type>
  <status>approved</status>
  <authorization_code>005645</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>live</mode>
  <timestamp>2025-02-26T16:01:48Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>5000</amount>
  <currency>USD</currency>
  <card_brand>master</card_brand>
  <card_number>555555...4444</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <payment_response>threeds</payment_response>
  <preauthorization>true</preauthorization>
  <preauthorization_expires_at>2025-03-29T16:01:48+00:00</preauthorization_expires_at>
  <preauthorization_total_amount>10000</preauthorization_total_amount>
  <captured>false</captured>
  <capturable_amount>8000</capturable_amount>
  <captured_amount>0</captured_amount>
  <reversed_amount>2000</reversed_amount>
  <reversible_amount>8000</reversible_amount>
  <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
  <sca_exemption_result>13</sca_exemption_result>
</payment_response>
```

---

## By date range

Date range based reconciliation allows you to fetch information for all payment transactions from a terminal within a given date range. The response is paginated, each request will return 100 entries max.

The URL for date range reconciling is:

`https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/by_date/TERMINAL-TOKEN`

Reconcile By Date Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/by_date/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <start_date>2014-01-01 09:20:00</start_date>
   <end_date>2014-01-31 21:30:00</end_date>
   <page>2</page>
 </reconcile>'
```

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required | yyyy-mm-dd hh:mm:ss | start of the requested date range (time is optional) |
| end_date | optional | yyyy-mm-dd hh:mm:ss | end of the requested date range (time is optional) |
| page | optional | integer the page within the paginated result, defaults to 1 | |

**Response:**

The attributes in the root node payment responses include information about the pagination of the response.

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_responses per_page="100" page="2" total_count="19" pages_count="7">
  <payment_response>
    <transaction_type>sale</transaction_type>
    <status>approved</status>
    <authorization_code>005645</authorization_code>
    <scheme_response_code>00</scheme_response_code>
    <response_code>00</response_code>
    <unique_id>130319cfb3bf65ff3c4a4045487b174f</unique_id>
    <transaction_id>EFBFBC7D-82CD-4375-9A69-15F19C88A134</transaction_id>
    <technical_message>Transaction successful!</technical_message>
    <message>Transaction successful!</message>
```

```
    <mode>live</mode>
    <timestamp>2014-01-03T15:04:00Z</timestamp>
    <descriptor>descriptor one</descriptor>
    <amount>500</amount>
    <currency>USD</currency>
    <card_brand>visa</card_brand>
    <card_number>420000...0000</card_number>
    <card_holder>John Doe</card_holder>
    <expiration_year>2020</expiration_year>
    <expiration_month>2</expiration_month>
    <sent_to_acquirer>true</sent_to_acquirer>
    <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
    <sca_exemption_result>13</sca_exemption_result>
  </payment_response>
  <payment_response>
    <transaction_type>sale</transaction_type>
    <status>approved</status>
    <authorization_code>638745</authorization_code>
    <scheme_response_code>00</scheme_response_code>
    <response_code>00</response_code>
    <unique_id>130319cfb3bf65ff3c4a4045487b173f</unique_id>
    <transaction_id>BBD7945B-BE57-4A14-A7FB-47F7AE928D95</transaction_id>
    <technical_message>Transaction successful!</technical_message>
    <message>Transaction successful!</message>
    <mode>live</mode>
    <timestamp>2014-01-05T15:04:00Z</timestamp>
    <descriptor>descriptor one</descriptor>
    <amount>500</amount>
    <currency>USD</currency>
    <card_brand>visa</card_brand>
    <card_number>420000...0000</card_number>
    <card_holder>John Doe</card_holder>
    <expiration_year>2020</expiration_year>
    <expiration_month>2</expiration_month>
    <sent_to_acquirer>true</sent_to_acquirer>
    <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
    <sca_exemption_result>13</sca_exemption_result>
  </payment_response>
  <payment_response>
    <transaction_type>sale</transaction_type>
    <status>approved</status>
    <authorization_code>226534</authorization_code>
    <scheme_response_code>00</scheme_response_code>
    <response_code>00</response_code>
    <unique_id>1e8a6f09253eb8b4f8c84c0d8803713e</unique_id>
    <transaction_id>_5041_2013041012_22_10_545</transaction_id>
    <technical_message>Transaction successful!</technical_message>
    <message>Transaction successful!</message>
    <mode>live</mode>
    <timestamp>2013-01-09T10:22:13Z</timestamp>
    <descriptor>test_descriptor</descriptor>
    <amount>50422</amount>
    <currency>EUR</currency>
    <card_brand>visa</card_brand>
    <card_number>420000...0000</card_number>
    <card_holder>John Doe</card_holder>
    <expiration_year>2020</expiration_year>
    <expiration_month>2</expiration_month>
    <sent_to_acquirer>true</sent_to_acquirer>
    <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
    <sca_exemption_result>13</sca_exemption_result>
  </payment_response>
  <payment_response>
    <transaction_type>sale3d</transaction_type>
    <status>pending_async</status>
    <unique_id>5dbdb4c677ee16b8fb1e43483164be2c</unique_id>
    <transaction_id>_6547_2013041012_23_08_470</transaction_id>
    <technical_message>Transaction successful!</technical_message>
    <message>Transaction successful!</message>
    <redirect_url>https://staging.gate.e-comprocessing.net/redirect/to_acquirer/5dbdb4c677</redirect_url>
    <mode>live</mode>
    <timestamp>2014-01-10T10:23:10Z</timestamp>
    <descriptor>test_descriptor</descriptor>
    <amount>100</amount>
    <currency>EUR</currency>
    <card_brand>visa</card_brand>
    <card_number>471110...0000</card_number>
    <card_holder>John Doe</card_holder>
    <expiration_year>2020</expiration_year>
    <expiration_month>2</expiration_month>
    <sent_to_acquirer>true</sent_to_acquirer>
    <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
    <sca_exemption_result>13</sca_exemption_result>
  </payment_response>
  <payment_response>
    <transaction_type>refund</transaction_type>
    <status>approved</status>
    <authorization_code>005645</authorization_code>
    <scheme_response_code>00</scheme_response_code>
    <response_code>00</response_code>
    <unique_id>44177a21403427eb96664a6d7e5d5d49</unique_id>
    <transaction_id>119643250547501c79d8296</transaction_id>
    <technical_message>Transaction successful!</technical_message>
    <message>Transaction successful!</message>
    <mode>test</mode>
    <timestamp>2014-01-30T14:21:48Z</timestamp>
    <descriptor>descriptor one</descriptor>
    <amount>9000</amount>
    <currency>USD</currency>
    <sent_to_acquirer>true</sent_to_acquirer>
    <reference_transaction_unique_id>44177a21403427eb96664a6d7e5d5d48</reference_transaction_unique_id>
    <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
    <sca_exemption_result>13</sca_exemption_result>
   </payment_response>
  ...
 </payment_responses>
```


| Name | Type | Description |
|---|---|---|
| @per_page | integer | number of entries per page |
| @page | integer | the current page |
| @total_count | integer | total number of all entries |
| @pages_count | integer | total number of pages |

# Payment Authorizations

**PAYMENT AUTHORIZATIONS API**

The Payment Authorizations API can be used to retrieve data about payment authorizations.

Payment Authorizations can be retrieved by date

**BY DATE RANGE**

Date range based retrieval allows you to fetch information for all payment authorizations for a given merchant within a given date range. Date range searches for payment authorizations by their creation date. The response is paginated, each request will return 100 entries max.

The URLs for date range payment authorizations retrieval are:

Production:

`https://gate.e-comprocessing.net/payment_authorizations/by_date`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/payment_authorizations/by_date`

Request

```
curl https://staging.gate.e-comprocessing.net/payment_authorizations/by_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_authorization_request>
   <start_date>2014-01-01</start_date>
   <end_date>2014-01-31</end_date>
   <auth_start_date>2014-01-01</auth_start_date>
   <auth_end_date>2014-01-31</auth_end_date>
   <externally_processed>external</externally_processed>
   <processing_type>all</processing_type>
   <page>1</page>
 </payment_authorization_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required* | yyyy-mm-dd | start of the requested date range |
| end_date | optional | yyyy-mm-dd | end of the requested date range |
| auth_start_date | required* | yyyy-mm-dd | start of the requested auth date range |
| auth_end_date | optional | yyyy-mm-dd | end of the requested auth date range |
| page | optional | integer | the page within the paginated result, defaults to 1 |
| per_page | optional | integer | Number of entities on page, defaults to 100 |
| externally_processed | optional | string(255) | Filters transactions by being externally processed: or being native to Genesis. Possible values include 'genesis', 'external', and 'all'. If flag not supplied, it defaults to 'genesis' |
| processing_type | optional | string(255) | Filters transactions by being card present or card not present. Possible values include 'card_present', 'card_not_present', and 'all'. If flag not supplied, it defaults to 'all'. |

`required* = conditionally required`

Note: One of **start_date** and **auth_start_date** is required.

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| merchant_number | string(20) | Merchant number |
| record_number | string | Unique identifier for the authorizations |
| card_number | string(19) | Masked card number |
| exp_date | string(4) | Expiry MMYY |
| currency | string(3) | Currency |
| amount | integer | Amount in minor currency unit, see Currency and Amount Handling for details |
| auth_code | string(6) | Authorization code |
| auth_date | string | Authorization date in date and time format |
| resp_code | string | Description of the response code |
| reversed | string(1) | Contains the character Y if the authorisation has been reversed else N |
| pos_entry_mode | string(3) | POS Entry Mode |
| voice | string(1) | Y if this is a voice authorisation, N otherwise |
| avs_result | string(1) | AVS Result |
| cvv2_result | string(1) | CVV2 Result |
| card_type | string(1) | Card Type code |
| ecom_type | string(5) | Indicator for electronic commerce transactions |
| eci_sli | string(2) | Visa card scheme ECI / MasterCard SLI |
| rrn | string(12) | Retrieval Reference Number |
| card_acceptor | string(40) | Card Acceptor Name and Location |
| mcc | string(4) | Merchant category code |
| trace | integer | Trace number. All messages belonging the same transaction have to have the same trace value |
| type | string(1) | Transaction Type |
| card_sub_type | string(8) | Card Subtype |
| terminal_id | string(15) | Terminal ID |
| unique_tran_id | string(36) | Unique Transaction Identifier |

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_authorizations_responses per_page="100" page="1" total_count="2" pages_count="1">
  <payment_authorizations_response>
    <xml_root>payment_authorization_response</xml_root>
    <merchant_number>605000000002029</merchant_number>
    <record_number>33202414090</record_number>
    <card_number>465944******7359</card_number>
    <exp_date/>
    <currency>EUR</currency>
    <amount>1234</amount>
    <auth_code>15056</auth_code>
    <auth_date>2019-06-10 16:17:18 UTC</auth_date>
    <resp_code>Transaction approved</resp_code>
    <reversed>N</reversed>
    <pos_entry_mode>102</pos_entry_mode>
    <voice>N</voice>
    <avs_result/>
    <cvv2_result>M</cvv2_result>
    <card_type>M</card_type>
    <ecom_type/>
    <eci_sli>21</eci_sli>
    <rrn>916116130233</rrn>
    <card_acceptor>test.bg                12629314     BG</card_acceptor>
    <mcc>7995</mcc>
    <trace>1130233</trace>
    <type/>
    <card_sub_type/>
    <terminal_id>53b1f5eacc9edd</terminal_id>
    <unique_tran_id/>
  </payment_authorizations_response>
  <payment_authorizations_response>
    <merchant_number>605000000002029</merchant_number>
    <record_number>33202414090</record_number>
    <card_number>465944******7359</card_number>
    <exp_date/>
    <currency>EUR</currency>
    <amount>1234</amount>
    <auth_code>15056</auth_code>
    <auth_date>2019-06-10 16:17:18 UTC</auth_date>
    <resp_code>Transaction approved</resp_code>
    <reversed>N</reversed>
    <pos_entry_mode>102</pos_entry_mode>
    <voice>N</voice>
    <avs_result/>
    <cvv2_result>M</cvv2_result>
    <card_type>M</card_type>
    <ecom_type/>
    <eci_sli>21</eci_sli>
    <rrn>916116130233</rrn>
    <card_acceptor>test.bg                12629314     BG</card_acceptor>
    <mcc>7995</mcc>
    <trace>1130233</trace>
    <type/>
    <card_sub_type/>
    <terminal_id>53b1f5eacc9edd</terminal_id>
    <unique_tran_id/>
  </payment_authorizations_response>
</payment_authorizations_responses>
```

The attributes in the root node payment authorization responses includes information about the pagination of the response.

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| @per_page | integer | number of entries per page |
| @page | integer | the current page |
| @total_count | integer | total number of all entries |
| @pages_count | integer | total number of pages |

# Processed Transactions

#### PROCESSED TRANSACTION API

The Processed Transaction API can be used to retrieve data about processed transactions.

#### SINGLE PROCESSED TRANSACTION

Single processed transaction retrieval allows to get a certain processed transaction by its ARN or by passing its unique ID.

Request

```
curl https://staging.gate.e-comprocessing.net/processed_transactions \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<processed_transaction_request>
  <arn>74537604221431003881865</arn>
</processed_transaction_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/processed_transactions \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<processed_transaction_request>
  <unique_id>53b1f5eacc9e4d3a3afebb4e993fe962</unique_id>
</processed_transaction_request>'
```

The URLs for the single processed transaction API are:

Production:

`https://gate.e-comprocessing.net/processed_transactions`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/processed_transactions`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<processed_transaction_response>
  <merchant_number>1240000000006698</merchant_number>
  <batch_number>eMP</batch_number>
  <transaction_id>a1qf12e81eb23d0e00ffb85b1db7d152</transaction_id>
  <transaction_date>2019-09-01 16:43:02 UTC</transaction_date>
  <post_date>2019-09-01</post_date>
  <terminal_id>53b1f5eacc9eddda3afebb4e993fe962</terminal_id>
  <auth_code>09117B</auth_code>
  <currency>USD</currency>
  <amount>3690</amount>
  <merchant_transaction_reference>b76e9a54bdc99b3</merchant_transaction_reference>
  <arn>85301169244934771128812</arn>
  <card_brand>MC World Signia</card_brand>
  <card_number>420000...0000</card_number>
  <bin_country>124</bin_country>
  <service_type_desc>Credit Card</service_type_desc>
  <merchant_country>826</merchant_country>
  <area_of_event>Foreign - MASTER</area_of_event>
  <cross_rate>1</cross_rate>
  <card_scheme>Mastercard</card_scheme>
  <capture_method>ICC, contactless, no cvm</capture_method>
  <unique_id>b76e9a54bdc99b338068681727ed5e240000</unique_id>
  <type>purchase</type>
  <card_present>false</card_present>
  <deposit_slip_number>60506291293</deposit_slip_number>
  <batch_slip_number>60506282664</batch_slip_number>
  <fees>
    <fee>
      <type>Assessment fee</type>
      <amount>-0.74</amount>
      <currency>USD</currency>
      <charge_amount>-0.74</charge_amount>
      <charge_currency>USD</charge_currency>
    </fee>
  </fees>
</processed_transaction_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| merchant_number | string(20) | Merchant number |
| batch_number | string(23) | Batch number |
| transaction_id | string(255) | Merchant transaction ID. Returned only when the processed transaction is Card Not Present. |
| transaction_date | string(255) | Transaction date in date and time format |
| post_date | string(255) | Posting date |
| terminal_id | string(10) | Terminal ID |
| arn | string(23) | Acquirer reference number |
| merchant_transaction_reference | string(23) | Merchant's transaction reference number |
| card_scheme | string(16) | Descriptive text for the card scheme |
| capture_method | string(255) | Capture method identifying the type of the transaction |
| service_type_desc | string(25) | Indicates if transaction is a Debit or Credit transaction |
| card_brand | string(3) | Scheme card brand |
| card_number | string(13) | Masked card number |
| bin_country | string(3) | Issuing BIN ISO Country Code from Scheme BIN tables |
| merchant_country | string(3) | 3 digit ISO country code of the merchant country |
| area_of_event | string(19) | Area of event |
| currency | string(3) | Currency of transaction |
| cross_rate | float(11) | FX rate to convert from transaction account to merchant funding currency |
| auth_code | string(6) | Authorization code |
| unique_id | string(36) | Unique Transaction Identifier is generated at PoS before sent for authorisation or offline approval |
| card_present | boolean | Transaction is card present or card not present |
| deposit_slip_number | string(11) | Deposit Slip Number |
| batch_slip_number | string(11) | Batch Slip Number |
| type | string(65) | Transaction type for the related charge posted to account |
| **fees** | | |
| type | string(65) | Transaction type for the related charge posted to account |
| amount | float | Calculated charge amount in transaction in major currency unit, see Currency and Amount Handling for details. |
| currency | string(3) | Currency of transaction |
| charge_amount | float | Calculated charge amount in charge currency in major currency unit, see Currency and Amount Handling for details. |
| charge_currency | string(3) | Currency of charge |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<processed_transaction_response>
  <status>error</status>
  <code>465</code>
  <message>Processed transaction not found!</message>
  <technical_message>Processed transaction by the given criteria cannot be found!</technical_message>
</processed_transaction_response>
```

In case no processed transaction is found for the given ARN or unique ID, a corresponding XML response is as follows:

**BY DATE OR POST DATE RANGE**

Date range based processed transaction retrieval allows you to fetch information for all processed transactions for a given merchant within a given date range. Date range searches for processed transactions either by their creation or posting date. The response is paginated, each request will return 100 entries max.

The URLs for date range processed transaction retrieval are:

Production:

`https://gate.e-comprocessing.net/processed_transactions/by_date`

`https://gate.e-comprocessing.net/processed_transactions/by_post_date`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/processed_transactions/by_date`

`https://staging.gate.e-comprocessing.net/processed_transactions/by_post_date`

Request

```
curl https://staging.gate.e-comprocessing.net/processed_transactions/by_post_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <processed_transaction_request>
   <start_date>2014-01-01</start_date>
   <end_date>2014-01-31</end_date>
   <batch_number>2065063</batch_number>
   <batch_slip_number>366236636</batch_slip_number>
   <deposit_slip_number>224234433</deposit_slip_number>
   <externally_processed>external</externally_processed>
   <processing_type>all</processing_type>
   <page>1</page>
 </processed_transaction_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required | yyyy-mm-dd | start of the requested date range |
| end_date | optional | yyyy-mm-dd | end of the requested date range |
| page | optional | integer | the page within the paginated result, defaults to 1 |
| per_page | optional | integer | Number of entities on page, defaults to 100 |
| batch_number | optional | string(255) | Batch number of processed transactions (only for `by_post_date` API call) |
| batch_slip_number | optional | string(255) | Batch slip number of processed transactions (only for `by_post_date` API call) |
| deposit_slip_number | optional | string(255) | Deposit slip number of processed transactions (only for `by_post_date` API call) |
| externally_processed | optional | string(255) | Filters transactions by being externally processed or being native to Genesis. Possible values include 'genesis', 'external', and 'all'. If flag not supplied, it defaults to 'genesis' |
| processing_type | optional | string(255) | Filters transactions by being card present or card not present. Possible values include 'card_present', 'card_not_present', and 'all'. If flag not supplied, it defaults to 'all'. |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<processed_transaction_responses per_page="100" page="1" total_count="2" pages_count="1">
  <processed_transaction_response>
    <merchant_number>124000000006698</merchant_number>
    <batch_number>eMP</batch_number>
    <transaction_id>a1qf12e81eb23d0e00ffb85b1db7d152</transaction_id>
    <transaction_date>2019-09-01 16:43:02 UTC</transaction_date>
    <post_date>2019-09-01</post_date>
    <terminal_id>53b1f5eacc9eddda3afebb4e993fe962</terminal_id>
    <auth_code>09117B</auth_code>
    <currency>USD</currency>
    <amount>3690</amount>
    <merchant_transaction_reference>b76e9a54bdc99b3</merchant_transaction_reference>
    <card_brand>MC World Signia</card_brand>
    <card_number>420000...0000</card_number>
    <bin_country>124</bin_country>
    <service_type_desc>Credit Card</service_type_desc>
    <merchant_country>826</merchant_country>
    <area_of_event>Foreign - MASTER</area_of_event>
    <cross_rate>1</cross_rate>
    <card_scheme>Mastercard</card_scheme>
    <capture_method>ICC, contactless, no cvm</capture_method>
    <unique_id>b76e9a54bdc99b338068681727ed5e240000</unique_id>
    <type>purchase</type>
    <card_present>false</card_present>
    <deposit_slip_number>60506291293</deposit_slip_number>
    <batch_slip_number>60506282664</batch_slip_number>
    <arn>85301169244934771128812</arn>
  </processed_transaction_response>
  <processed_transaction_response>
    <merchant_number>124000000006698</merchant_number>
    <batch_number>eMP</batch_number>
    <transaction_id>a1qf12e81eb23d0e00ffb85b1db7d152</transaction_id>
    <transaction_date>2019-09-01 16:43:02 UTC</transaction_date>
    <post_date>2019-09-01</post_date>
    <terminal_id>53b1f5eacc9eddda3afebb4e993fe962</terminal_id>
    <auth_code>09117B</auth_code>
    <currency>USD</currency>
    <amount>3690</amount>
    <merchant_transaction_reference>b76e9a54bdc99b3</merchant_transaction_reference>
    <card_brand>MC World Signia</card_brand>
    <card_number>420000...0000</card_number>
    <bin_country>124</bin_country>
    <service_type_desc>Credit Card</service_type_desc>
    <merchant_country>826</merchant_country>
    <area_of_event>Foreign - MASTER</area_of_event>
    <cross_rate>1</cross_rate>
    <card_scheme>Mastercard</card_scheme>
    <capture_method>ICC, contactless, no cvm</capture_method>
    <unique_id>b76e9a54bdc99b338068681727ed5e240000</unique_id>
    <type>purchase</type>
    <card_present>false</card_present>
    <deposit_slip_number>60506291293</deposit_slip_number>
    <batch_slip_number>60506282664</batch_slip_number>
```

```
    <arn>85301169244934771128812</arn>
    <fees>
      <fee>
        <type>Assessment fee</type>
        <amount>-0.74</amount>
        <currency>USD</currency>
        <charge_amount>-0.74</charge_amount>
        <charge_currency>USD</charge_currency>
      </fee>
    </fees>
  </processed_transaction_response>
</processed_transaction_responses>
```

The attributes in the root node processed transaction responses includes information about the pagination of the response.

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| @per_page | integer | number of entries per page |
| @page | integer | the current page |
| @total_count | integer | total number of all entries |
| @pages_count | integer | total number of pages |

# Processed Batches

#### PROCESSED BATCHES API

The Processed Batches API can be used to retrieve data about processed batches.

#### BY DATE OR POST DATE RANGE

Date range based processed batch retrieval allows you to fetch information for all processed batches for a given merchant within a given date range. Date range searches for processed batchess by their posting date. The response is paginated, each request will return 100 entries max.

The URLs for date range processed batch retrieval are:

Production:

`https://gate.e-comprocessing.net/processed_batches/by_post_date`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/processed_batches/by_post_date`

Request

```
curl https://staging.gate.e-comprocessing.net/processed_batches/by_post_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <processed_batch_request>
   <start_date>2014-01-01</start_date>
   <end_date>2014-01-31</end_date>
   <batch_slip_number>3354345543</batch_slip_number>
   <page>1</page>
 </processed_batch_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required | yyyy-mm-dd | start of the requested date range |
| end_date | optional | yyyy-mm-dd | end of the requested date range |
| batch_slip_number | optional | string(255) | Batch slip number of processed transactions |
| page | optional | integer | the page within the paginated result, defaults to 1 |
| per_page | optional | integer | Number of entities on page, defaults to 100 |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <processed_batch_responses per_page="100" page="1" total_count="2" pages_count="1">
   <processed_batch_response>
     <merchant_number>105000000002227</merchant_number>
     <batch_number>eMP</batch_number>
     <batch_slip_number>94280612195</batch_slip_number>
     <batch_date>2019-10-24</batch_date>
     <post_date>2019-10-24</post_date>
     <terminal_id>53b1f5eacc9eddda3afebb4e993fe962</terminal_id>
     <transaction_type>Misc. DR transaction</transaction_type>
     <number_of_transactions>3</number_of_transactions>
     <currency>EUR</currency>
     <amount>370</amount>
   </processed_batch_response>
 </processed_batch_responses>
```

The attributes in the root node processed batch responses includes information about the pagination of the response.

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| per_page | integer | number of entries per page |
| page | integer | the current page |
| total_count | integer | total number of all entries |
| pages_count | integer | total number of pages |
| batch_number | string(23) | Batch number. |
| batch_slip_number | string(11) | Batch slip number of the transaction. |

| | | |
|---|---|---|
| batch_date | yyyy-mm-dd | Batch date. |
| post_date | yyyy-mm-dd | Posting date. |
| terminal_id | string(10) | Terminal ID. |
| transaction_type | string(28) | The type of the transaction. |
| number_of_transactions | integer | The number of the transactions. |
| currency | string(3) | The currency of the transactions. |
| amount | float | The amount of the entry in major currency unit, see Currency and Amount Handling for details. |

# Chargebacks

Chargebacks are a special type of transactions as they cannot be triggered by the merchant. Charge- backs occur if a customer disputes to an item of his credit card bill at his issuing bank and the bank requests a chargeback. In this case, the amount is automatically refunded to the customers cc account and deducted from your merchant account.

Customers who initiate chargebacks will automatically be blocked for future transactions. For details, please contact our Risk team.

You can also see a chargeback overview in the merchant console under the Risk Management menu.

#### CHARGEBACK REVERSALS

The reversals could be split into two types. These are the chargeback reversals, which appear when the a chargeback dispute is cancelled (withdrawn) by the consumer/issuer and the representments, which appear when the merchant or the acquirer disputes an already received chargeback. Both of these chargeback event types are handled properly and integrated into the whole process of chargeback dispute procedure.

#### CHARGEBACK NOTIFICATIONS

You now have the option to receive API and/or email notifications for each chargeback event that occurs - e.g. for first chargebacks, second chargebacks, and representments. Enable this feature by emailing the IT Support team at tech-support@e-comprocessing.com with the chargeback notification URL if needed.

The email notifications are sent to the merchant user with role ’admin’ which is configured for managing the merchant entity on the gateway platform. The API notifications are equal to Notification for asynchronous payments, please refer to the section Notification for asynchronous payments to understand how notifications work.

Chargeback Notification Example

```
?transaction_id=343d9040a671c45832ee5381860e2996
&terminal_token=f4266042a6131b666060beb75691341d78ee5b4f
&unique_id=57ff74d1ca8727f59f243de6d01ff027
&transaction_type=sale
&status=chargebacked
&signature=ab4348afa9830834df90069646e4ce66c39a5358
&amount=100
&event=chargeback
```

#### CHARGEBACK API

The Chargeback API can be used to retrieve data about chargebacks.

#### SINGLE CHARGEBACK

Single chargeback retrieval allows to get a certain chargeback by its ARN or by passing the unique ID of the original transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/chargebacks \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <chargeback_request>
   <arn>74537604221431003881865</arn>
 </chargeback_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/chargebacks \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <chargeback_request>
   <original_transaction_unique_id>53b1f5eacc9e4d3a3afebb4e993fe962</original_transaction_unique_id>
 </chargeback_request>'
```

#### LIST OF CHARGEBACKS

Retrieve a list of chargebacks by ARN or by passing the unique ID of the original transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/chargebacks \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <chargeback_request>
   <arn>74537604221431003881865</arn>
   <mode>list</mode>
 </chargeback_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/chargebacks \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <chargeback_request>
   <original_transaction_unique_id>53b1f5eacc9e4d3a3afebb4e993fe962</original_transaction_unique_id>
   <mode>list</mode>
 </chargeback_request>'
```

The URLs for single and list of chargebacks API are:

Production:

`https://gate.e-comprocessing.net/chargebacks`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/chargebacks`

Successful Response

Response


```xml
<?xml version="1.0" encoding="UTF-8"?>
<chargeback_response>
  <type>1st Chargeback</type>
  <post_date>2014-01-24</post_date>
  <reason_code>4855</reason_code>
  <merchant_number>443344323459841</merchant_number>
  <reason_description>Non-receipt of merchandise</reason_description>
  <authorization_code>811714</authorization_code>
  <batch_number>2093064</batch_number>
  <cnn>9002578764</cnn>
  <merchant_transaction_reference>b76e9a54bdc99b3</merchant_transaction_reference>
  <capture_method>SET/3D-SET authenticated</capture_method>
  <amount>-14625</amount>
  <currency>USD</currency>
  <chargeback_amount>300.0</chargeback_amount>
  <chargeback_currency>EUR</chargeback_currency>
  <chargeback_account_amount>185.99</chargeback_account_amount>
  <chargeback_account_currency>EUR</chargeback_account_currency>
  <merchant_funding_amount>1003.72</merchant_funding_amount>
  <merchant_funding_currency>EUR</merchant_funding_currency>
  <original_transaction_amount>148.0</original_transaction_amount>
  <original_transaction_currency>EUR</original_transaction_currency>
  <merchant_settlement_amount>148.0</merchant_settlement_amount>
  <merchant_settlement_currency>EUR</merchant_settlement_currency>
  <network_settlement_amount>148.0</network_settlement_amount>
  <network_settlement_currency>EUR</network_settlement_currency>
  <merchant_dba_name>hyperstech.com</merchant_dba_name>
  <original_type>Purchase</original_type>
  <original_post_date>2019-06-28</original_post_date>
  <original_transaction_date>2019-06-28</original_transaction_date>
  <original_slip>92572791484</original_slip>
  <item_slip_number>93778283100</item_slip_number>
  <card_number>554960******5069</card_number>
  <card_brand>master</card_brand>
  <customer_email>jonh_doe@example.com</customer_email>
  <customer_phone>3598851248512</customer_phone>
  <transaction_type>sale3d</transaction_type>
  <original_transaction_unique_id>f9634ec5e7dbe6ca3871974accb875cd</original_transaction_unique_id>
  <arn>74537604221431003881865</arn>
</chargeback_response>
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| type | string(255) | The chargeback type. See chargeback types for details |
| post_date | string(255) | The date of the chargeback |
| reason_code | string(255) | Reason code of the chargeback |
| merchant_number | string(255) | Merchant number |
| reason_description | string(255) | Reason description of the chargeback |
| authorization_code | string(255) | Authorization code of the chargeback’s transaction |
| batch_number | string(23) | The batch number is provided by the submitter of the original presentment |
| cnn | string(14) | Chargeback Control Number filled for chargebacks and representments; empty for transfer transactions. |
| merchant_transaction_reference | string(255) | Merchant's transaction reference number |
| capture_method | string(255) | Capture method |
| amount | integer | Amount reported in the currency of the acquirer’s dispute account i.e. as per the accountancy of the acquirer. The amount can be negative for types 1st chargeback, 2nd chargeback, transfer reversal, and positive for all other types, see Currency and Amount Handling for details. |
| currency | string(3) | Currency as accounted by the acquirer. See ISO 4217 |
| chargeback_amount | float | The amount in the chargeback’s transaction currency i.e. the way it has been reported by the issuer. See Currency and Amount Handling for details. |
| chargeback_currency | string(3) | The currency of the chargeback as reported by the issuer. See ISO 4217 |
| chargeback_account_amount | float | Scheme settlement amount of the transaction |
| chargeback_account_currency | string(3) | Scheme settlement currency of the transaction |
| merchant_funding_amount | float | Amount corresponding to the financial impact for the merchant as reported by the acquirer |
| merchant_funding_currency | string(3) | Currency corresponding to the financial impact for the merchant as reported by the acquirer |
| original_transaction_amount | float(18) | Amount of the original presentment in transaction currency, see Currency and Amount Handling for details. |
| original_transaction_currency | string(3) | Transaction currency of the original presentment |
| merchant_settlement_amount | float(18) | Amount settled with the merchant for the original presentment (that is, the amount posted to the merchant account), before the deduction of any charges, see Currency and Amount Handling for details. |
| merchant_settlement_currency | string(3) | Currency settled with the payment network for the presentment before the deduction of any charges. |
| network_settlement_amount | float(18) | Amount settled with the merchant for the original presentment (that is, the amount posted to the merchant account), before the deduction of any charges, see Currency and Amount Handling for details. |
| network_settlement_currency | string(3) | Currency settled with the payment network for the presentment before the deduction of any charges. |
| merchant_dba_name | string(25) | Merchant name in the transaction as cleared to the schemes (charge descriptor). |
| original_type | string(28) | Transaction type of the original presentment |
| original_post_date | date(8) | Original presentment posting date |
| original_transaction_date | date(8) | Transaction date of the original presentment |
| original_slip | string(11) | OmniPay internal slip number of the original presentment |
| item_slip_number | string(11) | OmniPay internal slip number of the original presentment |
| card_number | string(255) | Card number used for the chargeback’s transaction |
| card_brand | string(255) | Card brand of the card number |
| customer_email | string(255) | The email of the cardholder |

| | | |
|---|---|---|
| customer_phone | integer | The phone of the cardholder |
| transaction_type | string(255) | The type of the chargeback's transaction |
| original_transaction_unique_id | string(255) | The unique id of the chargeback's transaction |
| arn | string(255) | ARN of the chargeback's transaction |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<chargeback_response>
  <status>error</status>
  <code>470</code>
  <message>Chargeback not found!</message>
  <technical_message>Chargeback by the given criteria cannot be found!</technical_message>
</chargeback_response>
```

In case no chargeback is found for the given ARN or unique ID, a corresponding XML response is as follows:

**BY DATE RANGE**

Date range based chargeback retrieval allows you to fetch information for all chargebacks for a given merchant within a given date range. Date range searches for chargebacks by their posting date. Search option is chargeback retrieval by their import (creation) date. The response is paginated, each request will return 100 entries max.

The URLs for date range chargeback retrieval are:

Production:

`https://gate.e-comprocessing.net/chargebacks/by_date`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/chargebacks/by_date`

Request

```
curl https://staging.gate.e-comprocessing.net/chargebacks/by_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <chargeback_request>
   <start_date>2014-01-01</start_date>
   <end_date>2014-01-31</end_date>
   <externally_processed>external</externally_processed>
   <processing_type>all</processing_type>
   <page>1</page>
 </chargeback_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required | yyyy-mm-dd | start of the requested date range |
| end_date | optional | yyyy-mm-dd | end of the requested date range |
| import_date | optional | yyyy-mm-dd | date of import in our system. Spans from beggining until end of day. |
| page | optional | integer | the page within the paginated result, defaults to 1 |
| per_page | optional | integer | Number of entities on page, defaults to 100 |
| externally_processed | optional | string(255) | Filters chargebacks by being externally processed or being native to Genesis. Possible values include 'genesis', 'external', and 'all'. If flag not supplied, it defaults to 'genesis' |
| processing_type | optional | string(255) | Filters chargebacks by being card present or card not present. Possible values include 'card_present', 'card_not_present', and 'all'. If flag not supplied, it defaults to 'all'. |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<chargeback_responses per_page="100" page="1" total_count="2" pages_count="1">
  <chargeback_response>
    <type>1st Chargeback</type>
    <merchant_number>443344323459841</merchant_number>
    <post_date>2014-01-24</post_date>
    <reason_code>4855</reason_code>
    <reason_description>Non-receipt of merchandise</reason_description>
    <authorization_code>811714</authorization_code>
    <batch_number>2093064</batch_number>
    <cnn>9002578764</cnn>
    <merchant_transaction_reference>b76e9a54bdc99b3</merchant_transaction_reference>
    <capture_method>SET/3D-SET authenticated</capture_method>
    <amount>-14625</amount>
    <currency>USD</currency>
    <chargeback_amount>300.0</chargeback_amount>
    <chargeback_currency>EUR</chargeback_currency>
    <chargeback_account_amount>185.99</chargeback_account_amount>
    <chargeback_account_currency>EUR</chargeback_account_currency>
    <merchant_funding_amount>1003.72</merchant_funding_amount>
    <merchant_funding_currency>EUR</merchant_funding_currency>
    <original_transaction_amount>148.0</original_transaction_amount>
    <original_transaction_currency>EUR</original_transaction_currency>
    <merchant_settlement_amount>148.0</merchant_settlement_amount>
    <merchant_settlement_currency>EUR</merchant_settlement_currency>
    <network_settlement_amount>148.0</network_settlement_amount>
    <network_settlement_currency>EUR</network_settlement_currency>
    <merchant_dba_name>hyperstech.com</merchant_dba_name>
    <original_type>Purchase</original_type>
    <original_post_date>2019-06-28</original_post_date>
    <original_transaction_date>2019-06-28</original_transaction_date>
    <original_slip>92572791484</original_slip>
    <item_slip_number>93778283100</item_slip_number>
    <card_number>554960******5069</card_number>
    <card_brand>master</card_brand>
    <customer_email>jonh_doe@example.com</customer_email>
    <customer_phone>3598851248512</customer_phone>
    <transaction_type>sale3d</transaction_type>
    <original_transaction_unique_id>f9634ec5e7dbe6ca3871974accb875cd</original_transaction_unique_id>
    <arn>74537604221431003881865</arn>
  </chargeback_response>
  <chargeback_response>
```

```
    <type>2nd Chargeback</type>
    <merchant_number>443344323459841</merchant_number>
    <post_date>2014-01-27</post_date>
    <reason_code>4855</reason_code>
    <reason_description>Non-receipt of merchandise</reason_description>
    <authorization_code>811714</authorization_code>
    <batch_number>2093064</batch_number>
    <cnn>9002578764</cnn>
    <merchant_transaction_reference>b76e9a54bdc99b3</merchant_transaction_reference>
    <capture_method>SET/3D-SET authenticated</capture_method>
    <amount>3456</amount>
    <currency>USD</currency>
    <chargeback_amount>300.0</chargeback_amount>
    <chargeback_currency>EUR</chargeback_currency>
    <chargeback_account_amount>185.99</chargeback_account_amount>
    <chargeback_account_currency>EUR</chargeback_account_currency>
    <merchant_funding_amount>1003.72</merchant_funding_amount>
    <merchant_funding_currency>EUR</merchant_funding_currency>
    <original_transaction_amount>148.0</original_transaction_amount>
    <original_transaction_currency>EUR</original_transaction_currency>
    <merchant_settlement_amount>148.0</merchant_settlement_amount>
    <merchant_settlement_currency>EUR</merchant_settlement_currency>
    <network_settlement_amount>148.0</network_settlement_amount>
    <network_settlement_currency>EUR</network_settlement_currency>
    <merchant_dba_name>hyperstech.com</merchant_dba_name>
    <original_type>Purchase</original_type>
    <original_post_date>2019-06-28</original_post_date>
    <original_transaction_date>2019-06-28</original_transaction_date>
    <original_slip>92572791484</original_slip>
    <item_slip_number>93778283100</item_slip_number>
    <card_number>454360******5008</card_number>
    <card_brand>visa</card_brand>
    <customer_email>ivan@example.com</customer_email>
    <customer_phone>3598851248512</customer_phone>
    <transaction_type>sale</transaction_type>
    <original_transaction_unique_id>67fbebc172b743a164a3f3af3d010457</original_transaction_unique_id>
    <arn>74537604221431003881865</arn>
  </chargeback_response>
</chargeback_responses>
```


The attributes in the root node chargeback responses includes information about the pagination of the response.

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| @per_page | integer | number of entries per page |
| @page | integer | the current page |
| @total_count | integer | total number of all entries |
| @pages_count | integer | total number of pages |

**CHARGEBACK TYPES**

Chargebacks will have one of the following type:

| Type | Description |
|---|---|
| 1st Chargeback | The first stage of the dispute procedure raised by the issuer |
| 2nd Chargeback | Second stage of the dispute procedure raised by the issuer (MasterCard only) |
| 1st Chargeback Reversal | When the first chargeback is cancelled (withdrawn) by the issuer |
| 2nd Chargeback Reversal | When the second chargeback is cancelled (withdrawn) by the issuer (MasterCard only) |
| Transfer Reversal | An operation that sends the amount of the dispute to the merchant when the acquirer represents a chargeback |
| Re-presentment | Acquirer’s defend of the issuer’s (first) chargeback |
| Representment Reversal | The Representment initiated by the merchant is cancelled or rejected by the acquirer (the merchant's bank) due to an error or invalid reason for representment. The funds that were initially refunded to the merchant as a result of the re-presentment are reversed back to the issuer account (the cardholder's bank). |
| Chargeback Transfer to Merchant Hold Acc | The money is taken from merchant’s hold account in Omnipay and is sent to the issuer |
| Chargeback Transfer to Writeoff Account Acq | The acquirer is taking the loss for this chargeback |
| Chargeback Transfer to Paymnt Acct Retail | The money is taken from merchant’s account in Omnipay and are sent to the issuer (as per the chargeback rules). |
| Chargeback Transfer to Writeoff SP | The Service Provider is taking the loss for the chargeback. |

# Rapid Dispute Resolution

Rapid Dispute Resolution(RDR) is a kind of a pre-dispute program of VISA via VISA's recent acquisition of VERIFI. Its goal is to reduce chargebacks.

RDR is now part of VROL and all the issuers have to use it now prior to initiating a dispute. A merchant can enrol via Visa's VERIFI service (or another authorized reseller) and define rules for auto-liability acceptance with the platform.

When an RDR event occurs and the merchant accepts the liability then the cardholder will be automatically refunded. VISA will withdraw the amount from the acquirer account.

**RAPID DISPUTE RESOLUTION API**

The Rapid Dispute Resolution API can be used to retrieve data about Rapid Dispute Resolutions.

**SINGLE RAPID DISPUTE RESOLUTION**

Single RDR retrieval allows to get a certain RDR by its ARN or by passing the unique ID of the original transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/rapid_dispute_resolutions \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <rapid_dispute_resolution_request>
   <arn>74537604221431003881865</arn>
 </rapid_dispute_resolution_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/rapid_dispute_resolutions \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <rapid_dispute_resolution_request>
   <original_transaction_unique_id>53b1f5eacc9e4d3a3afebb4e993fe962</original_transaction_unique_id>
 </rapid_dispute_resolution_request>'
```

**LIST OF RAPID DISPUTE RESOLUTIONS**

Retrieve a list of rdrs by ARN or by passing the unique ID of the original transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/rapid_dispute_resolutions \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <rapid_dispute_resolution_request>
   <arn>74537604221431003881865</arn>
   <mode>list</mode>
 </rapid_dispute_resolution_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/rapid_dispute_resolutions \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <rapid_dispute_resolution_request>
   <original_transaction_unique_id>53b1f5eacc9e4d3a3afebb4e993fe962</original_transaction_unique_id>
   <mode>list</mode>
 </rapid_dispute_resolution_request>'
```

The URLs for single and list of Rapid Dispute Resolutions API are:

Production:

`https://gate.e-comprocessing.net/rapid_dispute_resolutions`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/rapid_dispute_resolutions`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <rapid_dispute_resolution_response>
   <type>RDR Pre-dispute</type>
   <post_date>2014-01-24</post_date>
   <reason_code>4855</reason_code>
   <reason_description>Non-receipt of merchandise</reason_description>
   <rdr_amount>300.0</rdr_amount>
   <rdr_currency>EUR</rdr_currency>
   <card_number>554960******5069</card_number>
   <arn>74537604221431003881865</arn>
   <card_brand>master</card_brand>
   <customer_email>jonh_doe@example.com</customer_email>
   <customer_phone>3598851248512</customer_phone>
   <transaction_type>sale3d</transaction_type>
   <original_transaction_unique_id>f9634ec5e7dbe6ca3871974accb875cd</original_transaction_unique_id>
   <merchant_funding_amount>200.0</merchant_funding_amount>
   <merchant_funding_currency>EUR</merchant_funding_currency>
   <merchant_transaction_reference>12345</merchant_transaction_reference>
 </rapid_dispute_resolution_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| type | string(255) | The rapid dispute resolution type. |
| post_date | string(255) | The date of the rapid dispute resolution |
| reason_code | string(255) | Reason code of the rapid dispute resolution |
| reason_description | string(255) | Reason description of the rapid dispute resolution |
| rdr_amount | float | The amount in the rapid dispute resolution’s transaction currency i.e. the way it has been reported by the issuer. See Currency and Amount Handling for details. |
| rdr_currency | string(3) | The currency of the rapid dispute resolution as reported by the issuer. See ISO 4217 |
| card_number | string(255) | Card number used for the rapid dispute resolution’s transaction |
| arn | string(255) | ARN of the rapid dispute resolution’s transaction |
| card_brand | string(255) | Card brand of the card number |
| customer_email | string(255) | The email of the cardholder |
| customer_phone | integer | The phone of the cardholder |
| transaction_type | string(255) | The type of the rapid dispute resolution’s transaction |
| original_transaction_unique_id | string(255) | The unique id of the rapid dispute resolution’s transaction |
| merchant_funding_amount | string(255) | Merchant's funding amount |
| merchant_funding_currency | string(255) | Merchant's funding currency |
| merchant_transaction_reference | string(255) | Merchant reference id |

In case no rapid dispute resolution is found for the given ARN or unique ID, the corresponding XML response is as follows:

Error Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <rapid_dispute_resolution_response>
```

```
  <status>error</status>
  <code>471</code>
  <message>Rapid Dispute Resolution not found!</message>
  <technical_message>Rapid Dispute Resolution by the given criteria cannot be found!</technical_message>
</rapid_dispute_resolution_response>
```

#### BY DATE RANGE

Date range based rapid dispute resolution retrieval allows you to fetch information for all rapid dispute resolutions for a given date range. Date range searches for rapid dispute resolutions by their posting date if start_date and an optional end_date is provided. Date range can also search by creation date if import_date parameter is specified in the request. The response is paginated, each request will return maximum 100 entries.

The URLs for date range rapid dispute resolution retrieval are:

Production:

`https://gate.e-comprocessing.net/rapid_dispute_resolutions/by_date`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/rapid_dispute_resolutions/by_date`

Request

```
curl https://staging.gate.e-comprocessing.net/rapid_dispute_resolutions/by_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <rapid_dispute_resolution_request>
  <start_date>2014-01-01</start_date>
  <end_date>2014-01-31</end_date>
  <externally_processed>external</externally_processed>
  <processing_type>all</processing_type>
  <page>1</page>
 </rapid_dispute_resolution_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required | yyyy-mm-dd | start of the requested date range |
| end_date | optional | yyyy-mm-dd | end of the requested date range |
| import_date | optional | yyyy-mm-dd | date of import in our system. Spans from beginning until end of day. |
| page | optional | integer | the page within the paginated result, defaults to 1 |
| per_page | optional | integer | Number of entities on page, defaults to 100 |
| externally_processed | optional | string(255) | Filters rapid dispute resolutions by being externally processed or being native to Genesis. Possible values include 'genesis', 'external', and 'all'. If flag not supplied, it defaults to 'genesis' |
| processing_type | optional | string(255) | Filters rapid dispute resolutions by being card present or card not present. Possible values include 'card_present', 'card_not_present', and 'all'. If flag not supplied, it defaults to 'all'. |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <rapid_dispute_resolution_responses per_page="100" page="1" total_count="2" pages_count="1">
  <rapid_dispute_resolution_response>
    <type>RDR Pre-dispute</type>
    <post_date>2014-01-24</post_date>
    <reason_code>4855</reason_code>
    <reason_description>Non-receipt of merchandise</reason_description>
    <rdr_amount>300.0</rdr_amount>
    <rdr_currency>EUR</rdr_currency>
    <card_number>554960******5069</card_number>
    <arn>74537604221431003881865</arn>
    <card_brand>master</card_brand>
    <customer_email>jonh_doe@example.com</customer_email>
    <customer_phone>3598851248512</customer_phone>
    <transaction_type>sale3d</transaction_type>
    <original_transaction_unique_id>f9634ec5e7dbe6ca3871974accb875cd</original_transaction_unique_id>
    <merchant_funding_amount>200.0</merchant_funding_amount>
    <merchant_funding_currency>EUR</merchant_funding_currency>
    <merchant_transaction_reference>12345</merchant_transaction_reference>
  </rapid_dispute_resolution_response>
  <rapid_dispute_resolution_response>
    <type>RDR Pre-dispute</type>
    <post_date>2014-01-27</post_date>
    <reason_code>4855</reason_code>
    <reason_description>Non-receipt of merchandise</reason_description>
    <rdr_amount>300.0</rdr_amount>
    <rdr_currency>EUR</rdr_currency>
    <card_number>454360******5008</card_number>
    <arn>74537604221431003881865</arn>
    <card_brand>visa</card_brand>
    <customer_email>ivan@example.com</customer_email>
    <customer_phone>3598851248512</customer_phone>
    <transaction_type>sale</transaction_type>
    <original_transaction_unique_id>67fbebc172b743a164a3f3af3d010457</original_transaction_unique_id>
    <merchant_funding_amount>200.0</merchant_funding_amount>
    <merchant_funding_currency>EUR</merchant_funding_currency>
    <merchant_transaction_reference>12345</merchant_transaction_reference>
    <amount>3456</amount>
  </rapid_dispute_resolution_response>
 </rapid_dispute_resolution_responses>
```

The attributes in the root node rapid dispute resolution responses includes information about the pagination of the response.

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| @per_page | integer | number of entries per page |
| @page | integer | the current page |
| @total_count | integer | total number of all entries |
| @pages_count | integer | total number of pages |

#### RAPID DISPUTE RESOLUTION TYPES

RDRs will have one of the following type:

| Status | Description |
|---|---|
| RDR Pre-dispute | RDR for pre-dispute |
| RDR Pre-dispute Reversal | RDR Reversal means that the funds already taken from the acquirer with a previously received RDR are reversed to the acquirer's account with Visa and the acquirer could respectively credit the merchant. |
| RDR Transfer to Payment Account Retail / RDR Transfer to Paymnt Acct Retail | The money is taken from merchant’s account in Omnipay and are sent to the issuer (as per the RDR rules). |
| RDR Transfer to Merchant Hold Account | The money is taken from merchant’s hold account in Omnipay and is sent to the issuer |
| RDR Transfer to Write-off Acq | The acquirer is taking the loss for the RDR |
| RDR Transfer to Write-off SP | The Service Provider is taking the loss for the RDR |

# Retrieval Requests

Retrieval requests are a special type of transactions as they cannot be triggered by the merchant. Retrieval requests occur if the issuer requests additional documentation for a transaction. Retrieval requests do not have financial implication, but they indicate the issuer doubts a given transaction and can initiate a chargeback.

For details, please contact our Risk team.

You can see also see a retrieval request overview in the merchant console under the Risk Management menu.

## Retrieval Request notifications

You now have the option to receive API and/or email notifications for each retrieval request event. Enable this feature by emailing the IT Support team at tech-support@e-comprocessing.com with the desired notification URL.

The email notifications are sent to the merchant user with role ‘admin’ which is configured for managing the merchant entity on the gateway platform. The API notifications are equal to Notification for asynchronous payments, please refer to the section Notification for asynchronous payments to understand how notifications work.

Retrieval Notification Example

```
?transaction_id=30450
&terminal_token=cd577214de104fa0dd9c28486b3c817fd08c89a6
&unique_id=5de39380bf7ac7e1fc31cbd7805dc0ec
&transaction_type=sale
&status=chargebacked
&signature=98676c91391094b823d1f521d06cc129195952f9
&amount=400
&currency=USD
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&reason_code=10
&reason_description=Dispute+Transaction
&post_date=2014-07-16
&arn=17b4646c093b025
&event=retrieval_request
```

## Retrieval Request API

The retrieval request API can be used to get info for retrieval requests.

## Single Retrieval Request

Single retrieval request retrieval allows to get a certain retrieval request by its ARN or by passing the unique ID of the original transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/retrieval_equests \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <retrieval_request_request>
   <arn>74537604221431003881865</arn>
 </retrieval_request_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/retrieval_equests \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <retrieval_request_request>
   <original_transaction_unique_id>53b1f5eacc9e4d3a3afebb4e993fe962</original_transaction_unique_id>
 </retrieval_request_request>'
```

## List of Retrieval Request

Retrieve a list of retrieval requests by ARN or by passing the unique ID of the original transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/retrieval_equests \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <retrieval_request_request>
   <arn>74537604221431003881865</arn>
   <retrieval_request_request>all</retrieval_request_request>
```

```
</retrieval_request_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/retrieval_equests \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<retrieval_request_request>
  <original_transaction_unique_id>53b1f5eacc9e4d3a3afebb4e993fe962</original_transaction_unique_id>
  <mode>list</mode>
</retrieval_request_request>'
```

The URLs for the single and list of retrieval requests API are:

Production:

`https://gate.e-comprocessing.net/retrieval_equests`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/retrieval_requests`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<retrieval_request_response>
  <type>Retrieval request</type>
  <arn>74537604221431003881865</arn>
  <post_date>2014-01-24</post_date>
  <reason_code>42</reason_code>
  <reason_description>Cardholder request</reason_description>
  <authorization_code>811714</authorization_code>
  <merchant_number>1240000000006698</merchant_number>
  <issuer_number>0000002884</issuer_number>
  <item_slip_number>93778283100</item_slip_number>
  <original_type>Purchase</original_type>
  <original_slip>92572791484</original_slip>
  <original_batch_number>2093064</original_batch_number>
  <description>Action Control ID=20140224019152</description>
  <fulfillment_date>2019-06-28</fulfillment_date>
  <original_post_date>2019-06-28</original_post_date>
  <original_transaction_date>2019-06-28</original_transaction_date>
  <original_transaction_amount>148.0</original_transaction_amount>
  <original_transaction_currency>EUR</original_transaction_currency>
  <merchant_settlement_amount>148.0</merchant_settlement_amount>
  <merchant_settlement_currency>EUR</merchant_settlement_currency>
  <network_settlement_amount>148.0</network_settlement_amount>
  <network_settlement_currency>EUR</network_settlement_currency>
  <card_number>554960******5069</card_number>
  <card_brand>master</card_brand>
  <customer_email>jonh_doe@example.com</customer_email>
  <customer_phone>3598851248512</customer_phone>
  <transaction_type>sale3d</transaction_type>
  <original_transaction_unique_id>f9634ec5e7dbe6ca3871974accb875cd</original_transaction_unique_id>
</retrieval_request_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| type | string(255) | The retrieval request type. See retrieval request types for details |
| arn | string(255) | ARN of the retrieval request’s transaction |
| post_date | string(255) | The date of the retrieval request |
| reason_code | string(255) | Reason code of the retrieval request |
| reason_description | string(255) | Reason description of the retrieval request |
| authorization_code | string(255) | Authorization code of the retrieval request’s transaction |
| merchant_number | string(20) | Merchant number |
| issuer_number | string(14) | Issuer reference number for the retrieval request |
| item_slip_number | string(11) | OmniPay internal slip number of the original presentment |
| original_type | string(28) | Transaction type of the original presentment |
| original_slip | string(11) | OmniPay internal slip number of the original presentment |
| original_batch_number | string(23) | The batch number is provided by the submitter of the original presentment |
| description | string(255) | Free-text note entered by the institution and associated with the fulfilment |
| fulfillment_date | date(8) | Date on which the retrieval request was fulfilled. Empty if not yet fulfilled |
| original_post_date | date(8) | Original presentment posting date |
| original_transaction_date | date(8) | Transaction date of the original presentment |
| original_transaction_amount | float | Amount of the original presentment in major currency unit, see Currency and Amount Handling for details. |
| original_transaction_currency | string(3) | Transaction currency of the original presentment |
| merchant_settlement_amount | float | Amount settled with the merchant for the original presentment (that is, the amount posted to the merchant account) in major currency unit, before the deduction of any charges, see Currency and Amount Handling for details. |
| merchant_settlement_currency | string(3) | Currency settled with the payment network for the presentment before the deduction of any charges. |
| network_settlement_amount | float | Amount settled with the merchant for the original presentment (that is, the amount posted to the merchant account) in major currency unit, before the deduction of any charges, see Currency and Amount Handling for details. |
| network_settlement_currency | string(3) | Currency settled with the payment network for the presentment before the deduction of any charges. |
| card_number | string(255) | Card number used for the retrieval request’s transaction |
| card_brand | string(255) | Card brand of the card number |
| customer_email | string(255) | The email of the cardholder |
| customer_phone | integer | The phone of the cardholder |

| | | |
|---|---|---|
| transaction_type | string(255) | The type of the retrieval request’s transaction |
| original_transaction_unique_id | string(255) | The unique id of the retrieval request’s transaction |
| arn | string(255) | ARN of the retrieval request’s transaction |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<retrieval_request_response>
  <status>error</status>
  <code>480</code>
  <message>Retrieval request not found!</message>
  <technical_message>Retrieval request by the given criteria cannot be found!</technical_message>
</retrieval_request_response>
```

In case no retrieval request is found with the given ARN or unique ID, a corresponding XML error response is received.

---

# By date range

Date range based retrieval request retrieval allows you to fetch information for all retrieval requests for a given merchant within a given date range. Date range searches for retrieval requests by their posting date. The response is paginated, each request will return 100 entries max.

The URLs for date range retrieval request retrieval are:

Production:

`https://gate.e-comprocessing.net/retrieval_requests/by_date`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/retrieval_requests/by_date`

Request

```
curl https://staging.gate.e-comprocessing.net/retrieval_requests/by_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<retrieval_request_request>
  <start_date>2014-01-01</start_date>
  <end_date>2014-01-31</end_date>
  <page>1</page>
</retrieval_request_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required | yyyy-mm-dd | start of the requested date range |
| end_date | optional | yyyy-mm-dd | end of the requested date range |
| import_date | optional | yyyy-mm-dd | date of import in our system. Spans from beggining until end of day. |
| page | optional | integer | the page within the paginated result, defaults to 1 |
| per_page | optional | integer | Number of entities on page, defaults to 100 |
| externally_processed | optional | string(255) | Filters retrieval requests by being externally processed or being native to Genesis. Possible values include 'genesis', 'external', and 'all'. If flag not supplied, it defaults to 'genesis' |
| processing_type | optional | string(255) | Filters retrieval requests by being card present or card not present. Possible values include 'card_present', 'card_not_present', and 'all'. If flag not supplied, it defaults to 'all'. |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<retrieval_request_responses per_page="100" page="1" total_count="2" pages_count="1">
  <retrieval_request_response>
    <type>Retrieval request</type>
    <post_date>2014-01-24</post_date>
    <reason_code>42</reason_code>
    <reason_description>Cardholder request</reason_description>
    <authorization_code>811714</authorization_code>
    <merchant_number>124000000006698</merchant_number>
    <issuer_number>0000002884</issuer_number>
    <item_slip_number>93778283100</item_slip_number>
    <original_type>Purchase</original_type>
    <original_slip>92572791484</original_slip>
    <original_batch_number>2093064</original_batch_number>
    <description>Action Control ID=20140224019152</description>
    <fulfillment_date>2019-06-28</fulfillment_date>
    <original_post_date>2019-06-28</original_post_date>
    <original_transaction_date>2019-06-28</original_transaction_date>
    <original_transaction_amount>148.0</original_transaction_amount>
    <original_transaction_currency>EUR</original_transaction_currency>
    <merchant_settlement_amount>148.0</merchant_settlement_amount>
    <merchant_settlement_currency>EUR</merchant_settlement_currency>
    <network_settlement_amount>148.0</network_settlement_amount>
    <network_settlement_currency>EUR</network_settlement_currency>
    <card_number>554960******5069</card_number>
    <card_brand>master</card_brand>
    <customer_email>jonh_doe@example.com</customer_email>
    <customer_phone>3598851248512</customer_phone>
    <transaction_type>sale3d</transaction_type>
    <original_transaction_unique_id>f9634ec5e7dbe6ca3871974accb875cd</original_transaction_unique_id>
    <arn>74537604221431003881865</arn>
  </retrieval_request_response>
  <retrieval_request_response>
    <type>Retrieval request</type>
    <post_date>2014-01-27</post_date>
    <reason_code>42</reason_code>
    <reason_description>Cardholder request</reason_description>
    <authorization_code>811714</authorization_code>
    <merchant_number>124000000006698</merchant_number>
    <issuer_number>0000002884</issuer_number>
    <item_slip_number>93778283100</item_slip_number>
```

```
    <original_type>Purchase</original_type>
    <original_slip>92572791484</original_slip>
    <original_batch_number>2093064</original_batch_number>
    <description>Action Control ID=20140224019152</description>
    <fulfillment_date>2019-06-28</fulfillment_date>
    <original_post_date>2019-06-28</original_post_date>
    <original_transaction_date>2019-06-28</original_transaction_date>
    <original_transaction_amount>148.0</original_transaction_amount>
    <original_transaction_currency>EUR</original_transaction_currency>
    <merchant_settlement_amount>148.0</merchant_settlement_amount>
    <merchant_settlement_currency>EUR</merchant_settlement_currency>
    <network_settlement_amount>148.0</network_settlement_amount>
    <network_settlement_currency>EUR</network_settlement_currency>
    <card_number>454360******5008</card_number>
    <card_brand>visa</card_brand>
    <customer_email>ivan@example.net</customer_email>
    <customer_phone>359885124851 2</customer_phone>
    <transaction_type>sale</transaction_type>
    <original_transaction_unique_id>67fbebc172b743a164a3f3af3d010457</original_transaction_unique_id>
    <arn>74537604221431003881865</arn>
  </retrieval_request_response>
</retrieval_request_responses>
```

The attributes in the root node **retrieval_request_responses** includes information about the pagination of the response.

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| @per_page | integer | number of entries per page |
| @page | integer | the current page |
| @total_count | integer | total number of all entries |
| @pages_count | integer | total number of pages |

# Funding Accounts

## Funding Accounts API

The Funding Accounts API can be used to retrieve data for funding accounts.

**BY POST DATE**

Fetch merchant related funding accounts by post date.

Request

```
curl https://staging.gate.e-comprocessing.net/funding_accounts/by_post_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<funding_accounts_request>
  <start_date>2020-01-01</start_date>
  <end_date>2020-01-31</end_date>
  <externally_processed>external</externally_processed>
  <processing_type>all</processing_type>
  <page>1</page>
  <per_page>50</per_page>
</funding_accounts_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required | yyyy-mm-dd | Start of the requested post date range |
| end_date | optional | yyyy-mm-dd | End of the requested post date range |
| page | optional | integer | The page within the paginated result, defaults to 1 |
| per_page | optional | integer | Number of entities on page, defaults to 100 |
| externally_processed | optional | string(255) | Filters transactions by being externally processed or being native to Genesis. Possible values include 'genesis', 'external', and 'all'. If flag not supplied, it defaults to 'genesis' |
| processing_type | optional | string(255) | Filters transactions by being card present or card not present. Possible values include 'card_present', 'card_not_present', and 'all'. If flag not supplied, it defaults to 'all'. |

`required* = conditionally required`

The URLs for the funding accounts API are:

Production:

`https://gate.e-comprocessing.net/funding_accounts/by_post_date`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/funding_accounts/by_post_date`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<funding_accounts_responses>
  <funding_account_response>
    <merchant_number>505702571465826</merchant_number>
    <account_number>123445</account_number>
    <account_type>Merchant Account</account_type>
    <post_date>2020-01-02</post_date>
    <transaction_type>Funding</transaction_type>
    <slips>2</slips>
    <reference_number>60506291293</reference_number>
    <reversal_indicator>N</reversal_indicator>
    <currency>USD</currency>
    <amount>10.00</amount>
    <account_amount_gross>100.00</account_amount_gross>
    <account_total_charges>0.00</account_total_charges>
    <account_amount_net>100.00</account_amount_net>
    <card_brand>visa</card_brand>
    <batch_number>ERV</batch_number>
    <value_date>2019-12-12</value_date>
```

```
    <transaction_category>Charges &amp;amp; fees</transaction_category>
    <local_currency>USD</local_currency>
    <local_amount_gross>10.00</local_amount_gross>
    <account_currency>USD</account_currency>
  </funding_account_response>
</funding_accounts_responses>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| merchant_number | string(20) | Merchant number. |
| account_number | string(11) | OmniPay Unique Identifier for the Merchant account. |
| account_type | string(20) | The Type of the Account. |
| post_date | string(255) | The posting date on which the item in question was posted to the account. |
| transaction_type | string(65) | Transaction type description. For adjustment transactions this may include some explanatory text. |
| slips | string(10) | The number of internal slip numbers covered by the item. |
| reference_number | string(11) | This is a Slip Number in the OmniPay system for the transaction. |
| reversal_indicator | string(1) | Reversal Indicator Y if this is a reversal else N. |
| currency | string(3) | Swift code for the transaction currency of the item in question. |
| amount | float | Transaction amount of the item in major currency unit, see Currency and Amount Handling for details. |
| account_amount_gross | float(18,2) | The gross amount posted to the account, in the account currency rounded to 2 decimal places, see Currency and Amount Handling for details. |
| account_total_charges | float(18,2) | Applicable charges for the item, expressed in the account currency rounded to 2 decimals places, see Currency and Amount Handling for details. |
| account_amount_net | float(18,2) | The net amount in the account currency, after taking into account any charges rounded to 2 decimal places, see Currency and Amount Handling for details. |
| card_brand | string(16) | The card scheme associated to the transaction. Can be empty. |
| batch_number | string(23) | The 80byte batch number used to submit the transaction. This is only available for Transaction Types: -Merchant Purchase Deposits. -Merchant Debits. |
| arn | string(23) | Acquirer Reference Number. This is only available for the transaction Type: 'Chargeback'. |
| addendum_acquirer_reference | string(25) | ARN created by transferring a transaction or by Misc Batch Input. |
| fee_sequence | string(3) | Fee Sequence Code for a transaction. Usually provided for adjustment transaction in general it is empty. |
| fee_description | string(19) | Fee Sequence Description per Fee Sequence Code. |
| funding_date | string(255) | Associated Funding Date, only relevant for transactions whose category is 008 - Payments. |
| value_date | string(255) | Value Date. |
| transaction_category_code | string(3) | Transaction Category Code. |
| transaction_category | string(32) | Description for the transaction category code. |
| client_number | string(8) | This is the OmniPay Client Number and is the link to the MP feed. |
| payment_status | string(70) | The field will only be filled for payments if the institution uses the funding warehouse functionality. It will be empty for other transactions. |
| bank_reference | string(20) | This field is filled only for specific institutions on payment transactions. |
| account_type_code | string(3) | Account Type Code. |
| related_slip | string(11) | This field has details of a related transaction. Depending of the transaction type this could be a link to another transaction within the FA feed (Ref_no field) or a TR (Item_no field) or a PB (BTCH_SLIP field). |
| transaction_type_code | string(3) | Transaction type code. |
| local_currency | string(3) | Swift code of the local currency. |
| local_amount_gross | string(21) | The gross amount posted to the account, in the local currency, see Currency and Amount Handling for details. |
| account_currency | string(3) | Account Currency. |

Invalid End Date

Request

```
curl https://staging.gate.e-comprocessing.net/funding_accounts/by_post_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <funding_accounts_request>
   <start_date>2020-01-01</start_date>
   <end_date>13-2020</end_date>
 </funding_accounts_request>'
```

Error Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <funding_accounts_response>
   <status>error</status>
   <code>340</code>
   <message>Please check input data for errors!</message>
   <technical_message>input_error: 'end_date' has invalid format</technical_message>
 </funding_accounts_response>
```

# Fraud reports

SAFE/TC40 reports contain information for transactions reported as fraud to MasterCard or VISA.

You can see a SAFE/TC40 reports overview in the merchant console under the Risk management menu.

## SAFE/TC40 API

The SAFE/TC40 API can be used to retrieve data about SAFE/TC40 reports.

## Fraud report codes

| Code | Type | Description |
|---|---|---|
| 0 | string(1) | Lost |
| 1 | string(1) | Stolen |
| 2 | string(1) | Card not received as issued (NRI) |
| 3 | string(1) | Fraudulent application |
| 4 | string(1) | Issuer-reported counterfeit |
| 5 | string(1) | Miscellaneous/Account takeover |
| 6 | string(1) | Fraudulent use of account number |
| 7 | string(1) | (U.S. only) used by ICS |
| 8 | string(1) | (U.S. only) used by ICS |
| 9 | string(1) | Acquirer-reported counterfeit' |

## Single SAFE/TC40 report

Single SAFE/TC40 retrieval allows to get a certain SAFE/TC40 by its ARN or by passing the unique ID of the original transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/fraud_reports \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <fraud_report_request>
   <arn>74537604221431003881865</arn>
 </fraud_report_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/fraud_reports \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <fraud_report_request>
   <original_transaction_unique_id>53b1f5eacc9e4d3a3afebb4e993fe962</original_transaction_unique_id>
 </fraud_report_request>'
```

## List of SAFE/TC40 report

Retrieve a list of SAFE/TC40 by ARN or by passing the unique ID of the original transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/fraud_reports \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <fraud_report_request>
   <fraud_report_request>arn,</fraud_report_request>
   <mode>list</mode>
 </fraud_report_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/fraud_reports \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <fraud_report_request>
   <original_transaction_unique_id>53b1f5eacc9e4d3a3afebb4e993fe962</original_transaction_unique_id>
   <mode>list</mode>
 </fraud_report_request>'
```

The URLs for the single and list of SAFE/TC40 API are:

Production:

`https://gate.e-comprocessing.net/fraud_reports`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/fraud_reports`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <fraud_report_response>
   <transaction_date>2014-01-24</transaction_date>
   <fraud_type_code>6</fraud_type_code>
   <amount>400</amount>
   <currency>GBP</currency>
   <card_number>554960******5069</card_number>
   <report_date>2014-05-07</report_date>
   <card_brand>master</card_brand>
```

```
    <customer_email>name@example.net</customer_email>
    <customer_phone>3598851248512</customer_phone>
    <transaction_type>sale3d</transaction_type>
    <original_merchant_transaction_id>123294sss</original_merchant_transaction_id>
    <original_transaction_unique_id>f9634ec5e7dbe6ca3871974accb875cd</original_transaction_unique_id>
    <arn>74537604221431003881865</arn>
  </fraud_report_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_date | yyyy-mm-dd | The date transaction is made |
| fraud_type_code | string(1) | Fraud report codes. See the codes here |
| amount | integer | Amount in minor units in cardholder account currency. Amount can be negative for: 1st chargeback, 2nd chargeback, transfer reversal and positive for all other types. |
| currency | string(3) | Currency of the cardholder account. See ISO 4217 |
| card_number | string(255) | Card number used for the chargeback's transaction |
| report_date | yyyy-mm-dd | The report entered date |
| card_brand | string(255) | Card brand of the card number |
| customer_email | string(255) | The email of the cardholder |
| customer_phone | integer | The phone of the cardholder |
| transaction_type | string(255) | The type of the chargeback's transaction |
| original_merchant_transaction_id | string(255) | Merchant reference id |
| original_transaction_unique_id | string(255) | The unique id of the chargeback's transaction |
| arn | string(255) | ARN of the chargeback's transaction |

Error Response

Response

```
  <?xml version="1.0" encoding="UTF-8"?>
  <fraud_report_response>
    <status>error</status>
    <code>490</code>
    <message>Mastercard Fraud Report not found!</message>
    <technical_message>Mastercard fraud report by the given criteria cannot be found!</technical_message>
  </fraud_report_response>
```

In case no SAFE/TC40 is found for the given ARN or unique ID, a corresponding XML response is as follows:

---

# By date range

Date range based SAFE/TC40 retrieval allows you to fetch information for all SAFE/TC40 reports for a given merchant within a given date range. Date range searches include:
- for SAFE/TC40 reports by their posting date.
- for SAFE/TC40 retrieval by their import (creation) date.
- for SAFE/TC40 retrieval by their fraud report date.
The response is paginated, each request will return 100 entries max.

The URLs for date range SAFE/TC40 retrieval are:

Production:

`https://gate.e-comprocessing.net/fraud_reports/by_date`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/fraud_reports/by_date`

Request

```
curl https://staging.gate.e-comprocessing.net/fraud_reports/by_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <fraud_report_request>
   <start_date>2014-01-01</start_date>
   <end_date>2014-01-31</end_date>
   <page>1</page>
 </fraud_report_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/fraud_reports/by_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <fraud_report_request>
   <import_date>2014-01-01</import_date>
   <page>1</page>
 </fraud_report_request>'
```

OR

Request

```
curl https://staging.gate.e-comprocessing.net/fraud_reports/by_date \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <fraud_report_request>
   <report_start_date>2014-01-01</report_start_date>
   <report_end_date>2014-01-31</report_end_date>
   <page>1</page>
 </fraud_report_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required* | yyyy-mm-dd | Start of the requested date range |
| end_date | optional | yyyy-mm-dd | End of the requested date range |
| import_date | required* | yyyy-mm-dd | Date of import in our system. Spans from beginning until end of day. |
| report_start_date | required* | yyyy-mm-dd | Start of the requested date range for the date when the fraud was reported |
| report_end_date | optional | yyyy-mm-dd | End of the requested date range for the date when the fraud was reported |
| page | optional | integer | The page within the paginated result, defaults to 1 |
| per_page | optional | integer | Number of entities on page, defaults to 100 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<fraud_report_responses per_page="100" page="1" total_count="2" pages_count="1">
  <fraud_report_response>
    <transaction_date>2014-01-24</transaction_date>
    <fraud_type_code>6</fraud_type_code>
    <amount>400</amount>
    <currency>GBP</currency>
    <card_number>554960******5069</card_number>
    <report_date>2014-05-07</report_date>
    <card_brand>master</card_brand>
    <customer_email>name@example.net</customer_email>
    <customer_phone>3598851248512</customer_phone>
    <transaction_type>sale3d</transaction_type>
    <original_merchant_transaction_id>123294sss</original_merchant_transaction_id>
    <original_transaction_unique_id>f9634ec5e7dbe6ca3871974accb875cd</original_transaction_unique_id>
    <arn>74537604221431003881865</arn>
  </fraud_report_response>
  <fraud_report_response>
    <transaction_date>2014-01-27</transaction_date>
    <fraud_type_code>6</fraud_type_code>
    <amount>350</amount>
    <currency>GBP</currency>
    <card_number>454360******5008</card_number>
    <report_date>2014-05-27</report_date>
    <card_brand>visa</card_brand>
    <customer_email>name@example.net</customer_email>
    <customer_phone>3598851248512</customer_phone>
    <transaction_type>sale</transaction_type>
    <original_merchant_transaction_id>123294sss</original_merchant_transaction_id>
    <original_transaction_unique_id>67fbebc172b743a164a3f3af3d010457</original_transaction_unique_id>
    <arn>74537604221431003881865</arn>
  </fraud_report_response>
</fraud_report_responses>
```

**Conditionally required attributes mean that, you need to either send start/end_date, import_date or report_start/end_date.**

The attributes in the root node fraud report responses includes information about the pagination of the response.

# Blacklists

With the Blacklist API you can check if a certain credit card is blacklisted within the gateway. If a terminal token is not passed, the merchant and global PAN blacklists will be checked for the given card number. If a terminal token is passed, the terminal, its merchant, and the global PAN blacklists will be checked for blacklist matches.

The URLs for the Blacklist API are:

Production: `https://gate.e-comprocessing.net/blacklists`

Staging (for integration): `https://staging.gate.e-comprocessing.net/blacklists`

## Invoking a Request

A transaction is invoked via HTTPS POST, parameters are passed as XML with UTF-8 encoding.

```
<?xml version="1.0" encoding="UTF-8"?>
<blacklist_request>
  <card_number>4200000000000000</card_number>
  <terminal_token>abd30ed00f88f838c5d233cbb62b6da0b69267b4</terminal_token>
</blacklist_request>
```

| Parameter | Required | Format | Description |
|---|---|---|---|
| card_number | required | int(13..16) | the credit card number to be checked |
| terminal_token | optional | string(40) | the terminal token |

## Response

**Successful response:**

```
<?xml version="1.0" encoding="UTF-8"?>
<blacklist_response>
  <blacklisted>true</blacklisted>
</blacklist_response>
```

| Name | Type | Description |
|---|---|---|
| blacklisted | boolean | credit card number is blacklisted or not |

**Error response:**

```
<?xml version="1.0" encoding="UTF-8"?>
<blacklist_response>
  <code>350</code>
  <message>Invalid XML: No close tag for /blacklist_request</message>
```

```
</blacklist_response>
```

| Name | Type | Description |
|---|---|---|
| code | integer | error code of the error that occurred |
| message | string | info about the error |

# Asynchronous Transactions and Notifications

## Asynchronous Transactions

3D-Secure transactions can be either processed **asynchronously** or **synchronously** depending on the 3DSv2 authentication flow that will be reached (For more information, go to the 3DSv2-Authentication flows). Other types of transactions that are processed asynchronously, are payments that require the end-user to complete the payment using the **redirect_url** that is returned from the API within the synchronous response. Such transaction types are **Sofort**, **iDeal** Online Banking ePayments and others (For more information, check the documentation per transaction type and the available API responses - either synchronous or asynchronous response).

This means that the final result of the transaction will not be available immediately and the status is pending async. Once the transaction has reached a final status, a Notification is sent to the merchant.

> Whenever the status is **pending_async** the transaction gets processed asynchronously, and a Notification is sent to the merchant once the transaction has reached final state.

**Overview**

| Transaction type | async? |
|---|---|
| Authorize | never |
| Authorize3d | always |
| Sale | never |
| Sale3d | always |
| Capture | never |
| Refund | never |
| Async Refund | always |
| Void | never |
| InitRecurringSale | never |
| InitRecurringSale3D | always |
| RecurringSale | never |
| Credit | never |
| BitPay Sale | always |
| BitPay Refund | always |
| iDeal | always |
| Sofort | always |

## Notifications

For asynchronous payments, a notification is always sent, either to the **notification_url** provided within the payment transaction or to the one configured in the merchant account.

The payment gateway can be configured to also send a notification after each synchronous payment transaction. The notification is sent to the **notification_url** which is configured per merchant.

Also see 3-D Secure Transactions and Notification for asynchronous payments.

The format of the notification in both cases is the same.

Notification Example

```
?transaction_id=82803B4C-70CC-43BD-8B21-FD0395285B40
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=sale3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=approved
&amount=500
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&consumer_id=123456
&token=ee946db8-d7db-4bb7-b608-b65b153e127d
&eci=05
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&scheme_response_code=00
&scheme_transaction_identifier=MCS267BG0
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_settlement_date=1103
&reason_for_not_honoring_exemption=8A01
&sca_exemption_result=13
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
```

Sofort Notification Example

```
?transaction_id=82803B4C-70CC-43BD-8B21-FD0395285B40
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=sofort
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=approved
&amount=500
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&funds_status=sent
&account_holder=name+surname
&bank_account_number=DE89370400440532013000
&bank_identifier_code=GENODETT488
```

**Parameters**

| Name | Type | Description |
|---|---|---|
| transaction_id | string | merchant generated transaction id |
| unique_id | string | unique id generated by Genesis |
| transaction_type | string | transaction type for the transaction eg: sale3d |
| terminal_token | string | the terminal token as used in the processing url |
| status | string | status of the payment transaction |
| amount | string | amount of the payment transaction. If the transaction is partially approved, this is the partially approved amount. Check Partial Approvals for details |
| partial_approval | string | If the transaction is partially approved, this is set to 'true'. Check Partial Approvals for details |
| signature | string | the signature of the notification, should be used to verify the notification was sent by Genesis |
| funds_status | string | funds status of transaction *present only when the transaction has funds status |
| account_holder | string | account Holder of transaction's bank account. *present only when the transaction has account holder |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| token | string(36) | Plain-text token value. See Tokenize |
| eci | string | See Electronic Commerce Indicator as returned from the MPI for details |
| event | string | The event that caused the notification |
| rc_code | string | The reason code for the event |
| rc_description | string | The reason description for the event |
| avs_response_code | string | Generated by the card network on trying to match the billing ad- dress when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string | Card Verification Value response code. Optional, returned only if acquirer supports it. |
| reference_transaction_unique_id | string | The unique id generated by Genesis, identifies the reference transaction if present. |
| authorization code | string | A code returned by some acquirers to indicate that a card payment has been authorized. |
| retrieval_reference_number | string | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| scheme_response_code | string | The response code returned from the schemes. |
| recurring_advice_code | string | An additional response code returned from the schemes. Specifies if the transaction can be retried in case of failure. |
| recurring_advice_text | string | The text representation of the recurring advice code. |
| threeds_authentication_flow | string | Identifies the concrete 3DS authentication flow that the transaction has gone through. It will be available in the notification only if the consumer has finished the 3DS authentication with the issuer. The available values for 3DSv2 are frictionless and challenge. |
| threeds_method_status | string | Identifies the status of the 3DS-Method in the scope of 3DSv2 authentication protocol. The possible values are **required**, **in_progress** and **completed**. For more details about the 3DS-Method submission, go to the 3DSv2 authentication flows. |
| threeds_target_protocol_version | string(1) | Identifies the 3DS protocol that has been enforced. The possible values are **2**. |
| threeds_concrete_protocol_version | string(1) | Identifies the concrete 3DS protocol version that the transaction has gone through. The possible values are **2**. |
| threeds_protocol_sub_version | string(1) | Identifies the 3DS protocol sub-version that the transaction has gone through. The possible values are **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**. |
| threeds_authentication_status_reason_code | string(2) | See 3DS Authentication Status Reason Codes for details. |
| scheme_transaction_identifier | string | The text representation of the scheme transaction identifier. |
| scheme_settlement_date | string | The text representation of the scheme settlement date. |
| card_brand | string* | The brand of the card used for the transaction. |
| card_number | string* | The card number of the card used for the transaction. |
| card_type | string* | The type of the card used for the transaction. |
| card_subtype | string* | The subtype of the card used for the transaction. |
| card_issuing_bank | string* | The card issuer. |
| card_holder | string* | The card holder. |
| expiration_year | string* | The expiration year of the card. |
| expiration_month | string* | The expiration month of the card. |
| status | string* | The transaction status. |
| customer_email | string* | The email of the customer. |
| customer_phone | string* | The phone of the customer. |
| first_name | string* | The first name of the customer. |
| last_name | string* | The last name of the customer. |
| address1 | string* | The address of the customer. |
| address2 | string* | The second line of address of the customer. |
| zip_code | string* | The zip code of the customer. |
| city | string* | The city of the customer. |
| state | string* | The state of the customer. |
| country | string* | The country of the customer. |
| arn acquirer_reference_number | string* | The unique number assigned to the card transaction as it moves through the payment flow. |
| bank_account_number | string* | The IBAN number of the customer. |
| bank_identifier_code | string* | The BIC of the customer bank. |
| currency | string* | The currency of the transaction. |

| | | |
|---|---|---|
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |
| scheme_transaction_link_id | string | The transaction unique identifier returned from the schemes. |
| scheme_pan_indicator | string | The account number indicator in scheme-tokenized transactions. See Account number indicator codes for details. |
| scheme_pan_tail | string | The tail of the PAN number associated with the token in scheme-tokenized transactions. Returned in conjunction with `scheme_pan_indicator`. |
| payment_account_reference | string | The payment account reference value returned from the schemes. |
| scheme_max_settlement_date | string | The final date an authorization could be captured which is returned from the card scheme. See Scheme Maximum Settlement Date for more details. |
| scheme_authentication_data_quality | boolean | The Authentication Data Quality Indicator returned by the card scheme is used to indicate whether a transaction meets the authentication data quality requirements. The value can be either `true` or `false`. |

`string* = This is an optional parameter.` Contact tech-support@e-comprocessing.com for more details

Please, be advised that the **threeds_method_status** will be available only in the scope of 3DSv2 and if only the ACS Provider has requested a 3DS-Method to be submitted. For more information about the available 3DSv2 authentication flows with or without 3DS-Method, go to the 3DSv2 authentication flows.

For more information about the 3DS notification params and notification examples, go to the **Notification** section for each of the 3DSv2 authentication flows.

Status will be either "declined", "approved" or "error", like shown in the states table.

The event parameter is added only for fraud transactions eg: chargeback, chargeback_reversal, representment, representment_reversal, second_chargeback or retrieval_request.

The signature is a mean of security to ensure that the gate is really the sender of the notification. It is generated by concatenating the unique id of the transaction with your API password and generating a Hash of the string using SHA algorithm:

`SHA Hash Hex of <unique_id><Your API password>`

Possible encryption algorithms:

- SHA-1 (by default)
- SHA-256
- SHA-512

To change the encryption algorithm please contact Tech Support.

**Notification signature examples**

| unique_id | API password | algorithm | signature |
|---|---|---|---|
| fc6c3c8c0219730c7a099eaa540f70dc | bogus | SHA-1 | 08d01ae1ebdc22b6a1a764257819bb26e9e94e8d |
| 130319cfb3bf65ff3c4a4045487b173e | test123 | SHA-256 | e4c5e70de4a5b00663122f0b902ff4bb73f4542354e3a1edecf24a038576596d |
| a459f8781f2fe14a6e787648c146be02 | secret | SHA-512 | 162528f9760c188076ca1694701f7827e4904f2f7c72179a9c493989e8ba2c73318f818d61a4685485296f95a4c0aba0d826890eeef618a78df6ba50f170da69 |

You can use the signature to verify the integrity of the notification, ensuring that it was really sent by the gate.

You must either use the signature to verify the notification's integrity or make a reconcile to check the final transaction status.

```
<?xml version="1.0" encoding="UTF-8"?>
<notification_echo>
    <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
</notification_echo>
```

When receiving the notification, you are required to render an xml page containing the transaction's unique id so that the gateway knows that you have accepted the notification. If the XML is not delivered, the notification is sent periodically until the XML is received.

# WPF

The WPF (Web Payment Form) is a customizable component of the Genesis payment gateway. It provides merchants with an intuitive user interface to easily process their customers' payments. Through a single point of integration, the merchant can offer his customers multiple payment methods instantly and since the WPF is hosted on the secure Genesis infrastructure, it is already PCI-DSS compliant.

## Workflow

In the example above the customer visits the merchant's website and does a checkout.

(1), the merchant initiates the payment on the Genesis payment gateway through a request to the WPF API, which carries the mandatory, initial payload (e.g. amount, currency, etc.). The response to this request (if successful) is a redirect URL, which the merchants hands over to the customer. Following this redirect URL

(2), the customer is then directed to the actual payment form (WPF), which gets served from the Genesis servers. Because the merchant has previously transmitted most of the relevant payment information, the form is prefilled with these values and the customer only needs to add personal data. The customer then selects one of the payment methods offered by the merchant and fills in his payment information

(3) (e.g. credit card data). Upon completion the customer is redirected back to the merchant

(4)[1]. After the payment has been processed and reached a final state the merchant is sent a notification

(5) to the notification url supplied in the initial create request (1). The merchant must either use the notification's signature to verify the payment's integrity or make a reconcile

(6) to check the final payment status. However, we urge all merchants to always do a reconcile.

`1 - Particularly to the return success url defined by the merchant in his initial request. If the customer has selected an asynchronous payment method, he is redirected to the MPI provider before this step`

> 3D secure WPF payments are always performed asynchronously. After submitting the web payment form, the customer is redirected to the MPI provider to enter his personal data. In the case when cardholder is not enrolled customer is redirected to the failure url.
>
> As with all other asynchronous payment transactions, the return-, success- and cancel-URLs are only meant to display a useful page/message to the customer. A redirect of the customer to one of these URLs never gives any form of indication of the payment's state. To find out whether the payment has gone through or not the merchant must always wait for the notification or (even better) do a reconcile.
>
> Please also note that there is no specific order in which notification and redirect will occur (that means that the notification may also arrive before the customer's redirect).

---

## Express Checkout

**APPLE PAY**

When `Apple Pay` transaction type is sent in the initial WPF create request, the consumer will have an option to pay using enabled debit or credit card in the Apple Wallet instead of adding its payment details to the form, step (2) from the main workflow.

Apple Pay is available on all iOS devices with a Secure Element — an industry-standard, certified chip designed to store payment information safely. In macOS, users must have an Apple Pay-capable iPhone or Apple Watch to authorize the payment, or a MacBook Pro with Touch ID.

> Apple Pay is supported in Safari only.

In order to do so, while the consumer is on the WPF form page, there should be a `Apple Pay` button:

After this, the consumer should select the `Apple Pay` button.

Then a pop-up window with its predefined card details will appear and the consumer will be able to select any of the predefined cards to complete the payment. In macOS, consumers will be asked to turn on the Bluetooth on the mobile phone and pair the devices in order to be able to fetch and display the registered cards in Apple Wallet.

Lastly, the consumer should complete the payment either using biometric authentication such as fingerprint, Face ID on his/her mobile phone or to follow additional instructions on the popup window and the payment will be automatically processed with the next step (3) from the main workflow.

All other steps are the same as in the main workflow.

**GOOGLE PAY**

When `Google Pay` transaction type is sent in the initial WPF create request, the consumer will have an option to pay using saved to its `Google` account debit or credit card instead of adding its payment details to the form, step (2) from the main workflow.

In order to do so while the consumer is on the WPF form page, there should be a `Google Pay` button:

After this, the consumer should select the `Pay with Google Pay` button.

Then a pop-up window with its predefined card details will appear and the consumer will be able to select any of the predefined cards or add a new payment option which will be saved to its `Google` account for next payments.

Lastly, the consumer should confirm the selected payment option and the payment will be automatically processed with the next step (3) from the main workflow.

All other steps are the same as in the main workflow.

# WPF API

**URLS**

**Create:**

The URL for the WPF API create method is:

`https://wpf.e-comprocessing.net/<locale>/wpf`

For the test system the URL is:

`https://staging.wpf.e-comprocessing.net/<locale>/wpf`

Note that if you do not submit one of the available locales, defaults to 'en' (English).

Check the WPF Internationalization (i18n) for details.

**Reconcile:**

The URL for the WPF API reconcile method is:

`https://wpf.e-comprocessing.net/wpf/reconcile`

For the test system the URL is:

`https://staging.wpf.e-comprocessing.net/wpf/reconcile`

**CREATE**

Web Payment Form API supports the 3DSv2 authentication protocol for the following transaction types Authorize3d Sale3d InitRecurringSale3d. For more information, please check the request parameters below and request examples on the right.

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <wpf_payment>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <description>You are about to buy 3 shoes at www.shoes.com!</description>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
   <return_pending_url>http://www.example.com/payment-pending.html</return_pending_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_id>123456</consumer_id>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <remember_card>true</remember_card>
   <lifetime>60</lifetime>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
   <transaction_types>
     <transaction_type moto="true" name="authorize" fx_rate_id="123" recurring_type="initial"/>
     <transaction_type crypto="true" name="sale" fx_rate_id="123" recurring_type="initial"/>
   </transaction_types>
   <business_attributes>
     <name_of_the_supplier>Best Furniture</name_of_the_supplier>
   </business_attributes>
   <pay_later>true</pay_later>
   <reminder_language>en</reminder_language>
   <reminders>
     <reminder>
       <channel>email</channel>
       <after>40</after>
     </reminder>
     <reminder>
       <channel>sms</channel>
       <after>10</after>
     </reminder>
   </reminders>
   <sca_params>
     <exemption>low_value</exemption>
   </sca_params>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
   </account_owner>
   <web_payment_form_id>1</web_payment_form_id>
 </wpf_payment>'
```

Funding Transaction Example

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <wpf_payment>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <description>You are about to buy 3 shoes at www.shoes.com!</description>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
   <return_pending_url>http://www.example.com/payment-pending.html</return_pending_url>
   <amount>100</amount>
   <currency>USD</currency>
   <consumer_id>123456</consumer_id>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <remember_card>true</remember_card>
   <lifetime>60</lifetime>
```

```
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
  <transaction_types>
    <transaction_type moto="true" name="authorize" fx_rate_id="123" recurring_type="initial"/>
    <transaction_type crypto="true" name="sale" fx_rate_id="123" recurring_type="initial"/>
  </transaction_types>
  <business_attributes>
    <name_of_the_supplier>Best Furniture</name_of_the_supplier>
  </business_attributes>
  <pay_later>true</pay_later>
  <reminder_language>en</reminder_language>
  <reminders>
    <reminder>
      <channel>email</channel>
      <after>40</after>
    </reminder>
    <reminder>
      <channel>sms</channel>
      <after>10</after>
    </reminder>
  </reminders>
  <sca_params>
    <exemption>low_value</exemption>
  </sca_params>
  <account_owner>
    <first_name>Travis</first_name>
    <middle_name>Joe</middle_name>
    <last_name>Pastrana</last_name>
  </account_owner>
  <web_payment_form_id>1</web_payment_form_id>
  <funding>
    <identifier_type>business_disbursement</identifier_type>
    <business_application_identifier>funds_disbursement</business_application_identifier>
    <receiver>
      <first_name>Hamza</first_name>
      <last_name>Arshad</last_name>
      <country>AF</country>
      <account_number>090078601</account_number>
      <account_number_type>iban</account_number_type>
      <address>Gulberg 21 street</address>
      <state>CN</state>
      <city>Lahore</city>
    </receiver>
    <sender>
      <name>John Wick</name>
      <country>BG</country>
      <reference_number>90078602</reference_number>
      <address>Meow street 5</address>
      <state>CN</state>
      <city>Columbia</city>
    </sender>
  </funding>
</wpf_payment>'
```

Web Payment Form With 3 D Sv2 Authentication Protocol Request

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <description>You are about to buy 3 shoes at www.shoes.com!</description>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
  <return_pending_url>http://www.example.com/payment-pending.html</return_pending_url>
  <amount>100</amount>
  <currency>USD</currency>
  <consumer_id>123456</consumer_id>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <remember_card>true</remember_card>
  <lifetime>60</lifetime>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
  <transaction_types>
    <transaction_type name="authorize3d"/>
    <transaction_type name="wechat"/>
  </transaction_types>
  <business_attributes>
    <name_of_the_supplier>Best Furniture</name_of_the_supplier>
  </business_attributes>
  <pay_later>true</pay_later>
  <reminder_language>en</reminder_language>
  <reminders>
    <reminder>
      <channel>email</channel>
      <after>40</after>
    </reminder>
    <reminder>
      <channel>sms</channel>
      <after>10</after>
    </reminder>
  </reminders>
  <sca_params>
    <exemption>low_value</exemption>
  </sca_params>
  <account_owner>
    <first_name>Travis</first_name>
    <middle_name>Joe</middle_name>
```

```
    <last_name>Pastrana</last_name>
  </account_owner>
  <web_payment_form_id>1</web_payment_form_id>
</wpf_payment>'
```

Web Payment Form With 3 D Sv2 Authentication Protocol Including Additional Optional 3 D Sv2 Attributes Request

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <description>You are about to buy 3 shoes at www.shoes.com!</description>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
  <return_pending_url>http://www.example.com/payment-pending.html</return_pending_url>
  <amount>100</amount>
  <currency>USD</currency>
  <consumer_id>123456</consumer_id>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <remember_card>true</remember_card>
  <lifetime>60</lifetime>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
  <transaction_types>
    <transaction_type name="init_recurring_sale3d"/>
    <transaction_type name="sofort"/>
  </transaction_types>
  <business_attributes>
    <name_of_the_supplier>Best Furniture</name_of_the_supplier>
  </business_attributes>
  <threeds_v2_params>
    <control>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>goods</category>
    </purchase>
    <recurring>
      <expiration_date>27-08-2025</expiration_date>
      <frequency>30</frequency>
    </recurring>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
  </threeds_v2_params>
  <pay_later>true</pay_later>
  <reminder_language>en</reminder_language>
  <reminders>
    <reminder>
      <channel>email</channel>
      <after>40</after>
    </reminder>
    <reminder>
      <channel>sms</channel>
      <after>10</after>
    </reminder>
  </reminders>
  <sca_params>
    <exemption>low_value</exemption>
  </sca_params>
  <account_owner>
    <first_name>Travis</first_name>
    <middle_name>Joe</middle_name>
    <last_name>Pastrana</last_name>
  </account_owner>
  <web_payment_form_id>1</web_payment_form_id>
</wpf_payment>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| usage | optional | string(255) | Description of the transaction for later use. |
| amount | required* | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required* | string(3) | Currency code in ISO 4217 |
| description | optional | string | a text describing the reason of the payment (e.g. "you’re buying concert tickets") |

| | | | |
|---|---|---|---|
| consumer_id | required* | string(10) | See Consumers and Tokenization. Saved cards will be listed for user to select |
| customer_email | required* | e-mail address | Must contain valid e-mail of customer |
| customer_phone | required* | string(32) | Must contain valid phone number of customer |
| notification_url | required* | url | URL at merchant where gateway sends outcome of transaction. |
| return_success_url | required* | url | URL where customer is sent to after successful payment |
| return_failure_url | required* | url | URL where customer is sent to after unsuccessful payment |
| return_cancel_url | required* | string | URL where customer is sent to when the customer cancels the payment process within the WPF |
| return_pending_url | optional | string | URL where customer is sent to when asynchronous payment is pending confirmation |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required* | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| neighborhood | optional | string(255) | Neighborhood |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **transaction_types** | required | | The transaction types that the merchant is willing to accept payments for |
| transaction_type | required | string | One of the available WPF transaction types. Multiple transaction type elements supported. Attribute ’name’ contains the transaction type name in question |
| **business_attributes** | required* | | Check business attributes section. |
| event_start_date | required* | dd-mm-yyyy | The date when event starts in format dd-mm-yyyy |
| event_end_date | required* | dd-mm-yyyy | The date when event ends in format dd-mm-yyyy |
| event_organizer_id | required* | string | |
| event_id | required* | string | |
| date_of_order | required* | dd-mm-yyyy | The date when order was placed in format dd-mm-yyyy |
| delivery_date | required* | dd-mm-yyyy | Date of the expected delivery in format dd-mm-yyyy |
| name_of_the_supplier | required* | string | |
| recurring_type | optional | string(255) | Specifies recurring type of the transaction, can be 'initial' or 'managed' when Sale or Sale3d or Authorize or Authorize3d is included in the **transaction_types**. Apple Pay and Google Pay transaction types supports 'initial' recurring type |
| recurring_category | optional | | Specifies whether the recurring transaction is a subscription(fixed amount, fixed intervals)or if it is a standing order(varying amount, fixed intervals). The allowed values are `subscription` and `standing_order`. The default value is `subscription` |
| **threeds_v2_params** | optional | | 3DSv2 async parameters. They must be submitted in order to use the 3DSv2 authentication protocol in asynchronous workflow |
| **control** | optional | | General params for preferences in authentication flow and providing device interface information. |
| challenge_window_size | optional | string | Identifies the size of the challenge window for the consumer. For more information, go to 3DSv2 control params |
| challenge_indicator | optional | string | The value has weight and might impact the decision whether a challenge will be required for the transaction or not. If not provided, it will be interpreted as **no_preference**. For more information, go to 3DSv2 control params |
| **purchase** | optional | | Purchase related params providing with additional information regarding the order. |
| category | optional | string | Optional for transactions to be processed through the 3DSv2 authentication protocol. |
| **recurring** | optional | | Additional optional recurring attributes when InitRecurringSale3d is included in the **transaction_types**. |
| expiration_date | optional | dd-mm-yyyy | A future date indicating the end date for any further subsequent transactions. For more information, go to 3DSv2 recurring params |
| frequency | optional | integer | Indicates the minimum number of days between subsequent transactions. An empty value indicates the payment frequency is not set.For more information, go to 3DSv2 recurring params |
| **merchant_risk** | optional | | Merchant risk assessment params. They are all optional, but recommended. |
| shipping_indicator | optional | string(16) | Indicator code that most accurately describes the shipping method for the cardholder specific transaction. If one or more items are included in the sale, use the Shipping Indicator code for the physical goods. If all digital goods, use the code that describes the most expensive item. Accepted values are: **same_as_billing**, **stored_address**, **verified_address**, **pick_up**, **digital_goods**, **travel**, **event_tickets**, **other**. |
| delivery_timeframe | optional | string(11) | Indicates the merchandise delivery timeframe. Accepted values are: **electronic**, **same_day**, **over_night**, **another_day**. |
| reorder_items_indicator | optional | string(10) | Indicates whether the cardholder is reordering previously purchased merchandise. Accepted values are: **first_time**, **reordered**. |

| | | | |
|---|---|---|---|
| pre_order_purchase_indicator | optional | string(21) | Indicates whether cardholder is placing an order for merchandise with a future-availability or release date. Accepted values are: **merchandise_available**, **future_availability**. |
| pre_order_date | optional | dd-mm-yyyy | For a pre-ordered purchase, the expected date that the merchandise will be available. |
| gift_card | optional | 'true' | Prepaid or gift card purchase. |
| gift_card_count | optional | integer | For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased. The value is limited to **99**. |
| **card_holder_account** | optional | | Cardholder account additional information. They are all optional, but recommended, because they have a significant impact on approval rates |
| creation_date | optional | dd-mm-yyyy | Date that the cardholder opened the account with the 3DS Requester. |
| update_indicator | optional | string(19) | Length of time since the cardholder's account information with the 3DS Requestor was last changed. Includes Billing or Shipping address, new payment account, or new user(s) added. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| last_change_date | optional | dd-mm-yyyy | Date that the cardholder's account with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment accout, or new user(s) added. |
| password_change_indicator | optional | string(18) | Length of time since the cardholder account with the 3DS Requestor had a password change or account reset. Accepted values are: **no_change**, **during_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| password_change_date | optional | dd-mm-yyyy | Date that cardholder's account with the 3DS Requestor had a password change or account reset. |
| shipping_address_usage_indicator | optional | string(19) | Indicates when the shipping address used for this transaction was first used with the 3DS Requestor. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| shipping_address_date_first_used | optional | dd-mm-yyyy | Date when the shipping address used for this transaction was first used with the 3DS Requestor. |
| transactions_activity_last_24_hours | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours. |
| transactions_activity_previous_year | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year. |
| provision_attempts_last_24_hours | optional | integer | Number of Add Card attempts in the last 24 hours. |
| purchases_count_last_6_months | optional | integer | Number of purchases with this cardholder account during the previous six months. |
| suspicious_activity_indicator | optional | string(22) | Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account. Accepted values are: **no_suspicious_observed**, **suspicious_observed**. |
| registration_indicator | optional | string(19) | Indicates the length of time that the payment account was enrolled in the cardholder's account with the 3DS Requester. Accepted values are: **guest_checkout**, **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| registration_date | optional | dd-mm-yyyy | Date that the payment account was enrolled in the cardholder's account with the 3DS Requestor. |
| remember_card | optional | "true" | See Tokenize. Offer the user the option to save cardholder details for future use (tokenize). |
| lifetime | optional | integer | number of minutes determining how long the WPF will be valid. Will be set to 30 minutes by default. Valid value ranges between 1 minute and 3 months given in minutes (131487) |
| **risk_params** | optional | | list of risk params as described in the Advanced risk management with RiskParams section |
| user_id (example) | optional | string | the customer's ID within the merchant's system (example) |
| session_id (example) | optional | string | the customer's session ID within the mer- chant's system (example) |
| pay_later | optional | "true" | Signifies whether the 'Pay Later' feature would be enabled on the WPF |
| reminder_language | optional | string | It must be a valid language abbreviation from the available WPF languages |
| **reminders** | optional | | Settings for reminders sending when using the 'Pay Later' feature. The number of the sent reminders would be exactly as sent or configured and delivery failures could be handled on demand. Also there will be no reminders sent if the WPF is already completed |
| **reminder** | optional | | Settings for a single reminder. Upto three reminders are allowed |
| channel | optional | string | Channel for sending WPF reminder. Valid values are 'email' and 'sms' |
| after | optional | integer | Number of minutes after WPF creation when the reminder should be sent. Valid value ranges between 1 minute and 31 days given in minutes |
| crypto | optional | "true" | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |
| gaming | optional | "true" | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| moto | optional | "true" | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| fx_rate_id | optional | integer | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| sca_preference | optional | "true" | Values 'true' or 'false'. Signifies whether to perform SCA on the transaction. At least one 3DS transaction type has to be submitted. Contact tech-support@e-comprocessing.com for more details |
| **sca_params** | optional | | SCA params |
| exemption | optional | string | Exemption for the Strong Customer Authentication. The allowed options are `low_value`, `low_risk` |
| **account_owner** | optional | | Account owner parameters related to account owner inquiry requests |
| first_name | optional | string(35) | Account owner's first name |
| middle_name | optional | string(35) | Account owner's middle name |
| last_name | optional | string(35) | Account owner's last name |
| web_payment_form_id | optional | | The unique ID of the the web payment form configuration to be displayed for the current payment. |
| **funding** | optional | | Funding Transaction Params |
| identifier_type | required* | string | This is only required in case of Mastercard. Type of Funding Transaction. Please check Identifier Types |
| business_application_identifier | required* | string | This is only required in case of VISA. Type of VISA Funding Transaction. Please check BAI |
| **receiver** | optional | | Funding Transaction Receiver details |
| first_name | required* | string | First name of the receiver |
| last_name | required* | string | Last name of the receiver |
| country | required* | string(2) | Country code in ISO 3166 |
| account_number | required* | string | Receiver account number. Mandatory for Mastercard. If Sender Reference Number is provided then it becomes optional for VISA |

| | | | |
|---|---|---|---|
| account_number_type | required* | string | This is only required in case of Mastercard. Receiver account number type. Please check Receiver Account Types |
| address | required* | string | Receiver address. Only required in case of VISA and Canadian cards |
| state | required* | string | Receiver state. Only required in case of VISA and Canadian cards |
| city | required* | string | Receiver city. Only required in case of VISA and Canadian cards |
| **sender** | optional | | Funding Transaction Sender details |
| name | required* | string | Sender name. Only required in case of VISA |
| reference_number | required* | string | Sender Reference Number. Only required in case of VISA. This is option if Receiver Account Number is provided |
| country | required* | string | Sender country. Only required in case of VISA. If not provided billing address is used |
| address | required* | string | Sender address. Only required in case of VISA. If not provided billing address is used |
| state | required* | string | Sender state. Only required in case of VISA and Canadian card. If not provided billing address is used |
| city | required* | string | Sender city. Only required in case of VISA. If not provided billing address is used |
| **dynamic_descriptor_params** | optional | | |
| merchant_name | optional | string(25) | Allows to dynamically override the charge descriptor |
| merchant_city | optional | string(13) | Allows to dynamically override the merchant phone number |
| sub_merchant_id | optional | string(15) | Allows to dynamically override the sub-merchant ID. |
| merchant_country | optional | string(3) | Allows to dynamically override the merchant country. |
| merchant_state | optional | string(3) | Allows to dynamically override the merchant subdivision code. |
| merchant_zip_code | optional | string(10) | Allows to dynamically override the merchant zip/postal code. Required for VISA OCT transactions with Australian and Canadian card bins. |
| merchant_address | optional | string(48) | Allows to dynamically override the merchant address. |
| merchant_url | optional | string(60) | Allows to dynamically override the merchant URL |
| merchant_phone | optional | string(16) | Allows to dynamically override the merchant phone number. |
| merchant_service_city | optional | string(13) | Allows to dynamically override the merchant service city. |
| merchant_service_country | optional | string(3) | Allows to dynamically override the merchant service country. |
| merchant_service_state | optional | string(3) | Allows to dynamically override the merchant service subdivision code. |
| merchant_service_zip_code | optional | string(10) | Allows to dynamically override the merchant service zip/postal code. |
| merchant_service_phone | optional | string(16) | Allows to dynamically override the merchant service phone number. |
| merchant_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant geographic coordinates. |
| merchant_service_geo_coordinates | optional | string(20) | Allows to dynamically override the merchant service geographic coordinates. |

`required* = conditionally required`

The ‘amount’ and ‘currency’ parameters are **NOT** required when all of the submitted transaction type(s) have no financial impact on consumer's account (i.e. ‘account_verification’)

The required business attributes must be provided with the WPF request if you are submitting at least one transaction type that supports business attributes

If the ‘web_payment_form_id’ parameter is not provided in the request the merchant's default web payment form configuration will be used

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <transaction_type>wpf_create</transaction_type>
  <status>new</status>
  <mode>live</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <consumer_id>123456</consumer_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <technical_message>Transaction successful!</technical_message>
  <message>Transaction successful!</message>
  <redirect_url>https://staging.wpf.e-comprocessing.net/en/payment/c7e32c1e9d1</redirect_url>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <amount>100</amount>
  <currency>USD</currency>
</wpf_payment>
```

Successful With Optional Invalid Transactions For Amount Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <consumer_id>123456</consumer_id>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <amount>100</amount>
  <currency>USD</currency>
  <redirect_url>https://staging.wpf.e-comprocessing.net/en/payment/c7e32c1e9d1</redirect_url>
  <invalid_transactions_for_amount>alipay</invalid_transactions_for_amount>
</wpf_payment>
```

Successful With Funds Status Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <consumer_id>123456</consumer_id>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <amount>100</amount>
  <currency>USD</currency>
  <redirect_url>https://staging.wpf.e-comprocessing.net/en/payment/c7e32c1e9d1</redirect_url>
  <funds_status>WAITING</funds_status>
</wpf_payment>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the WPF transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |
| redirect_url | url | URL where user has to be redirected to complete payment process. Contains the locale in which the web payment form will be rendered by default. See WPF Internationalization (i18n) |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| invalid_transactions_for_amount | string* | list of comma separated transactions for which amount is not within the allowed limit *present only when at least for one transaction type an amount is invalid, check Currency and Amount Handling for details |
| funds_status | string* | funds status of transaction *present only when the transaction has funds status |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <transaction_type>wpf_create</transaction_type>
  <status>error</status>
  <mode>live</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <code>330</code>
  <technical_message>Unknown system error. Please contact support.</technical_message>
  <message>Transaction failed, please contact support!</message>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <amount>100</amount>
  <currency>USD</currency>
</wpf_payment>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the WPF transaction, see states |
| mode | string(4) | Mode of the transaction's terminal, can be **test** or **live** |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

**NOTIFICATION**

WPF Notifications are sent once the WPF payment has reached a final state and are transmitted via HTTP POST (application/x-www-form-urlencoded) with the following parameters:

> Apart from the workflow described above, the Genesis system will also use the notification url endpoint to send notifications to the merchant if a WPF payment is chargebacked.

Example Notification for Successful Frictionless 3dsv2

```
signature=c5219b3d385e74496b2b48a5497b347e102849f10eacd25b062f823b
&payment_transaction_transaction_type=sale3d
&payment_transaction_terminal_token=e9fd7a957845450fb7ab9dccb498b6e1f6e1e3aa
&payment_transaction_unique_id=bad08183a9ec545daf0f24c48361aa10
&payment_transaction_amount=500
&wpf_transaction_id=mtid20110408144716113553696 2
&wpf_status=approved
&wpf_unique_id=26aa150ee68b1b2d6758a0e6c44fce4c
&consumer_id=123456
&payment_transaction_token=ee946db8-d7db-4bb7-b608-b65b153e127d
&notification_type=wpf
&eci=05
&payment_transaction_avs_response_code=5I
&payment_transaction_avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&payment_transaction_cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_response_code=00
&scheme_transaction_identifier=MCS267BG0
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_settlement_date=1103
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&reason_for_not_honoring_exemption=8A01
&sca_exemption_result=13
&threeds_authentication_flow=frictionless
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

Example Notification for Successful Frictionless 3dsv2 With 3ds Method

```
signature=c5219b3d385e74496b2b48a5497b347e102849f10eacd25b062f823b
```

```
&payment_transaction_transaction_type=sale3d
&payment_transaction_terminal_token=e9fd7a957845450fb7ab9dccb498b6e1f6e1e3aa
&payment_transaction_unique_id=bad08183a9ec545daf0f24c48361aa10
&payment_transaction_amount=500
&wpf_transaction_id=mtid20110408144716113553696 2
&wpf_status=approved
&wpf_unique_id=26aa150ee68b1b2d6758a0e6c44fce4c
&consumer_id=123456
&payment_transaction_token=ee946db8-d7db-4bb7-b608-b65b153e127d
&notification_type=wpf
&eci=05
&payment_transaction_avs_response_code=5I
&payment_transaction_avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&payment_transaction_cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_response_code=00
&scheme_transaction_identifier=MCS267BG0
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_settlement_date=1103
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&reason_for_not_honoring_exemption=8A01
&sca_exemption_result=13
&threeds_authentication_flow=frictionless
&threeds_method_status=completed
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

Example Notification for Successful Challenge 3dsv2

```
signature=c5219b3d385e74496b2b48a5497b347e102849f10eacd25b062f823b
&payment_transaction_transaction_type=sale3d
&payment_transaction_terminal_token=e9fd7a957845450fb7ab9dccb498b6e1f6e1e3aa
&payment_transaction_unique_id=bad08183a9ec545daf0f24c48361aa10
&payment_transaction_amount=500
&wpf_transaction_id=mtid20110408144716113553696 2
&wpf_status=approved
&wpf_unique_id=26aa150ee68b1b2d6758a0e6c44fce4c
&consumer_id=123456
&payment_transaction_token=ee946db8-d7db-4bb7-b608-b65b153e127d
&notification_type=wpf
&eci=05
&payment_transaction_avs_response_code=5I
&payment_transaction_avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&payment_transaction_cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_response_code=00
&scheme_transaction_identifier=MCS267BG0
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_settlement_date=1103
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&reason_for_not_honoring_exemption=8A01
&sca_exemption_result=13
&threeds_authentication_flow=challenge
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

Example Notification for Successful Challenge 3dsv2 With 3ds Method

```
signature=c5219b3d385e74496b2b48a5497b347e102849f10eacd25b062f823b
&payment_transaction_transaction_type=sale3d
&payment_transaction_terminal_token=e9fd7a957845450fb7ab9dccb498b6e1f6e1e3aa
&payment_transaction_unique_id=bad08183a9ec545daf0f24c48361aa10
&payment_transaction_amount=500
&wpf_transaction_id=mtid20110408144716113553696 2
&wpf_status=approved
&wpf_unique_id=26aa150ee68b1b2d6758a0e6c44fce4c
&consumer_id=123456
&payment_transaction_token=ee946db8-d7db-4bb7-b608-b65b153e127d
&notification_type=wpf
&eci=05
&payment_transaction_avs_response_code=5I
&payment_transaction_avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&payment_transaction_cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_response_code=00
&scheme_transaction_identifier=MCS267BG0
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_settlement_date=1103
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&reason_for_not_honoring_exemption=8A01
&sca_exemption_result=13
&threeds_authentication_flow=challenge
&threeds_method_status=completed
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

Example Notification for Sofort

```
?transaction_id=82803B4C-70CC-43BD-8B21-FD0395285B40
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=sofort
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=approved
&amount=500
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&funds_status=sent
&account_holder=name+surname
&bank_account_number=DE89370400440532013000
&bank_identifier_code=GENODETT488
```

| Name | Type | Description |
|---|---|---|
| signature | string | the signature of the notification, should be used to verify the the notification was sent by Genesis |
| payment_transaction_transaction_type | string | transaction type for the transaction eg: sale3d |

| | | |
|---|---|---|
| payment_transaction_terminal_token | string | the terminal token as used in the processing url |
| payment_transaction_unique_id | string | unique id generated by Genesis |
| payment_transaction_amount | string | Amount of the payment transaction. If the transaction is partially approved, this is the partially approved amount. Check Partial Approvals for details |
| payment_transaction_partial_approval | string | If the transaction is partially approved, this is set to 'true'. Check Partial Approvals for details |
| wpf_transaction_id | string | merchant generated tranaction id |
| wpf_status | string | status of the payment transaction |
| wpf_unique_id | string | unique id generated by Genesis, required for WPF payment reconciliation |
| consumer_id | string(10) | Consumer unique reference. See Consumers |
| payment_transaction_token | string(36) | Plain-text token value. See Tokenize |
| notification_type | string | constant value "wpf" |
| eci | string | See Electronic Commerce Indicator as returned from the MPI for details |
| payment_transaction_avs_response_code | string | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| payment_transaction_avs_response_text | string | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| payment_transaction_cvv_result_code | string | Card Verification Value response code. Optional, returned only if acquirer supports it. |
| authorization code | string | A code returned by some acquirers to indicate that a card payment has been authorized. |
| retrieval_reference_number | string | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| funds_status | string | Funds status of transaction *present only when the transaction has funds status |
| account_holder | string | Account Holder of transaction's bank account. *present only when the transaction has account holder |
| scheme_response_code | string | The response code returned from the schemes. |
| recurring_advice_code | string | An additional response code returned from the schemes. Specifies if the transaction can be retried in case of failure. |
| recurring_advice_text | string | The text representation of the recurring advice code. |
| scheme_transaction_identifier | string | The text representation of the scheme transaction identifier. |
| scheme_settlement_date | string | The text representation of the scheme settlement date. |
| card_brand | string* | The brand of the card used for the transaction. |
| card_number | string* | The card number of the card used for the transaction. |
| card_type | string* | The type of the card used for the transaction. |
| card_subtype | string* | The subtype of the card used for the transaction. |
| card_issuing_bank | string* | The card issuer. |
| card_holder | string* | The card holder. |
| expiration_year | string* | The expiration year of the card. |
| expiration_month | string* | The expiration month of the card. |
| status | string* | The transaction status. |
| customer_email | string* | The email of the customer. |
| customer_phone | string* | The phone of the customer. |
| first_name | string* | The first name of the customer. |
| last_name | string* | The last name of the customer. |
| address1 | string* | The address of the customer. |
| address2 | string* | The second line of address of the customer. |
| zip_code | string* | The zip code of the customer. |
| city | string* | The city of the customer. |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |
| state | string* | The state of the customer. |
| country | string* | The country of the customer. |
| arn acquirer_reference_number | string* | The unique number assigned to the card transaction as it moves through the payment flow. |
| bank_account_number | string* | The IBAN number of the customer. |
| currency | string* | The currency of the transaction. |
| bank_identifier_code | string* | The BIC of the customer bank. |
| scheme_transaction_link_id | string | The transaction unique identifier returned from the schemes. |
| scheme_pan_indicator | string | The account number indicator in scheme-tokenized transactions. See Account number indicator codes for details. |
| scheme_pan_tail | string | The tail of the PAN number associated with the token in scheme-tokenized transactions. Returned in conjunction with `scheme_pan_indicator`. |
| scheme_max_settlement_date | string | The final date an authorization could be captured which is returned from the card scheme. See Scheme Maximum Settlement Date for more details. |
| scheme_authentication_data_quality | boolean | The Authentication Data Quality Indicator returned by the card scheme is used to indicate whether a transaction meets the authentication data quality requirements. The value can be either `true` or `false`. |
| payment_account_reference | string | The Payment Account Reference value returned from the schemes. |

`string* = This is an optional parameter.` Contact tech-support@e-comprocessing.com for more details

**3DS Attributes**

| Name | Type | Description |
|---|---|---|
| threeds_authentication_flow | string | Identifies the concrete 3DS authentication flow that the transaction has gone through. It will be available in the notification only if the consumer has finished the 3DS authentication with the issuer. The available values for 3DSv2 are frictionless and challenge. |

| | | |
|---|---|---|
| threeds_method_status | string | Identifies the status of the 3DS-Method in the scope of 3DSv2 authentication protocol. The possible values are **required**, **in_progress** and **completed**. |
| threeds_target_protocol_version | string(1) | Identifies the 3DS protocol that has been enforced. The possible values are **2**. |
| threeds_concrete_protocol_version | string(1) | Identifies the concrete 3DS protocol version that the transaction has gone through. The possible values are **2**. |
| threeds_protocol_sub_version | string(1) | Identifies the 3DS protocol sub-version that the transaction has gone through. The possible values are **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**. |
| threeds_authentication_status_reason_code | string(2) | See Status Reason Code for details. |

Status will be one of the following: approved, declined, error, timeout, pending, refunded, voided, chargebacked, chargeback reversed, represented, representment reversed and second chargebacked.

The signature is a mean of security to ensure that the gate is really the sender of the notification. It is generated by concatenating the unique id of the payment with your API password and generating a SHA-512 Hash (Hex) of the string:

```
SHA-512 Hash Hex of <wpf_unique_id><Your Merchant API password>
```

**Notification signature examples**

| unique id | API password | signature |
|---|---|---|
| 26aa150ee68b1b2d6758a0e6c44fce4c | 50fd87e65eb415f42fb5af4c9cf497662e00b785 | c5219b3d385e74496b2b48a549 |
| 3f760162ef57a829011e5e2379b3fa17 | 50fd87e65eb415f42fb5af4c9cf497662e00b785 | 14519d0db2f7f8f407efccc9b099 |

You must either use the signature to verify the notification's integrity or make a reconcile to check the final transaction status.

```
<?xml version="1.0" encoding="UTF-8"?>
<notification_echo>
  <wpf_unique_id>3f760162ef57a829011e5e2379b3fa17</wpf_unique_id>
</notification_echo>
```

When receiving the notification, you are required to render an xml page containing the transaction's unique id so that the gateway knows that you have accepted the notification. If the XML is not delivered, the notification is sent periodically until the XML is received.

#### RECONCILE

Reconcile can be used to retrieve data about a payment. This can be useful if you want to retrieve information about a payment whose status is timeout, which returned an error or has changed eg. has beed chargebacked.

Reconcile requests are handled exactly like transaction requests via XML. To a large degree, the WPF Reconcile follows the notions of the standard processing Reconcile API.

```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_reconcile>
  <unique_id>26aa150ee68b1b2d6758a0e6c44fce4c</unique_id>
</wpf_reconcile>
```

| Parameter | Required | Format | Description |
|---|---|---|---|
| unique_id | required | string(32) | unique id as returned by the create request |

Please note that the response may include multiple payment transaction records, since a WPF payment is a container class for multiple payment transactions. Card brand and card number will be available in response only for card transaction types.

Please note, a new response xml node will appear for 3DS transactions depending on the 3DS authentication flow and protocol.

Example Reconcile Response XML for Successful Frictionless 3dsv2:

```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <status>approved</status>
  <unique_id>26aa150ee68b1b2d6758a0e6c44fce4c</unique_id>
  <transaction_id>mtid20110408144716113553696 2</transaction_id>
  <consumer_id>123456</consumer_id>
  <timestamp>2011-04-08T14:46:27Z</timestamp>
  <amount>5000</amount>
  <currency>USD</currency>
  <usage>Shopify Electronic Transaction</usage>
  <description>You are about to buy shoes from Shopify</description>
  <card_brand>visa</card_brand>
  <card_number>401200...0085</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <card_holder>John Doe</card_holder>
  <expiration_year>2020</expiration_year>
  <expiration_month>2</expiration_month>
  <payment_transaction>
    <status>approved</status>
    <authorization_code>345678</authorization_code>
    <scheme_response_code>00</scheme_response_code>
    <avs_response_code>5I</avs_response_code>
    <avs_response_text>Response+provided+by+issuer+processor%3B+Address+information+not+verified</avs_response_text>
    <cvv_result_code>S</cvv_result_code>
    <response_code>00</response_code>
    <transaction_type>sale3d</transaction_type>
    <unique_id>bad08183a9ec545daf0f24c48361aa10</unique_id>
    <transaction_id>mtid20110408144716113553696 2</transaction_id>
    <arn>74537605259536043849425</arn>
    <terminal_token>e9fd7a957845450fb7ab9dccb498b6e1f6e1e3aa</terminal_token>
    <mode>test</mode>
    <timestamp>2011-04-08T14:46:40Z</timestamp>
    <descriptor>e-comprocessing.net/bogus +49123456789</descriptor>
    <bank_account_number>Bank Account Number</bank_account_number>
    <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
    <amount>4900</amount>
    <partial_approval>true</partial_approval>
    <currency>USD</currency>
    <customer_email>john.doe@example.com</customer_email>
    <customer_phone>+11234567890</customer_phone>
    <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
    <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
    <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
    <scheme_pan_indicator>V</scheme_pan_indicator>
    <scheme_pan_tail>0013</scheme_pan_tail>
    <scheme_settlement_date>0227</scheme_settlement_date>
    <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
    <scheme_authentication_data_quality>false</scheme_authentication_data_quality>
    <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
    <sca_exemption_result>13</sca_exemption_result>
    <technical_message>TESTMODE: No real money will be transferred!</technical_message>
    <message>TESTMODE: No real money will be transferred!</message>
      <billing_address>
        <first_name>John</first_name>
```

```
        <last_name>Doe</last_name>
        <address1>32, Doestreet</address1>
        <address2></address2>
        <zip_code>12345</zip_code>
        <city>New York</city>
      <state>NY</state>
      <country>US</country>
    </billing_address>
    <threeds>
      <authentication_flow>frictionless</authentication_flow>
      <protocol>
        <target_version>2</target_version>
        <concrete_version>2</concrete_version>
      </protocol>
      <eci>05</eci>
    </threeds>
  </payment_transaction>
</wpf_payment>
```


Example Reconcile Response XML for Successful Challenge 3dsv2 With 3ds Method:


```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <status>approved</status>
  <unique_id>26aa150ee68b1b2d6758a0e6c44fce4c</unique_id>
  <transaction_id>mtid20110408144716113553696 2</transaction_id>
  <consumer_id>123456</consumer_id>
  <timestamp>2011-04-08T14:46:27Z</timestamp>
  <amount>5000</amount>
  <currency>USD</currency>
  <usage>Shopify Electronic Transaction</usage>
  <description>You are about to buy shoes from Shopify</description>
  <card_brand>visa</card_brand>
  <card_number>493873...0001</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <card_holder>John Doe</card_holder>
  <expiration_year>2020</expiration_year>
  <expiration_month>2</expiration_month>
  <payment_transaction>
    <status>approved</status>
    <authorization_code>345678</authorization_code>
    <scheme_response_code>00</scheme_response_code>
    <avs_response_code>5I</avs_response_code>
    <avs_response_text>Response+provided+by+issuer+processor%3B+Address+information+not+verified</avs_response_text>
    <cvv_result_code>S</cvv_result_code>
    <response_code>00</response_code>
    <transaction_type>sale3d</transaction_type>
    <unique_id>bad08183a9ec545daf0f24c48361aa10</unique_id>
    <transaction_id>mtid20110408144716113553696 2</transaction_id>
    <arn>74537605259536043849425</arn>
    <terminal_token>e9fd7a957845450fb7ab9dccb498b6e1f6e1e3aa</terminal_token>
    <mode>test</mode>
    <timestamp>2011-04-08T14:46:40Z</timestamp>
    <descriptor>e-comprocessing.net/bogus +49123456789</descriptor>
    <bank_account_number>Bank Account Number</bank_account_number>
    <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
    <amount>4900</amount>
    <partial_approval>true</partial_approval>
    <currency>USD</currency>
    <customer_email>john.doe@example.com</customer_email>
    <customer_phone>+11234567890</customer_phone>
    <payment_account_reference>50019P9LBXOLHN9G7QMU5VN520YSY</payment_account_reference>
    <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
    <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
    <scheme_pan_indicator>V</scheme_pan_indicator>
    <scheme_pan_tail>0013</scheme_pan_tail>
    <scheme_settlement_date>0227</scheme_settlement_date>
    <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
    <scheme_authentication_data_quality>false</scheme_authentication_data_quality>
    <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
    <sca_exemption_result>13</sca_exemption_result>
    <technical_message>TESTMODE: No real money will be transferred!</technical_message>
    <message>TESTMODE: No real money will be transferred!</message>
      <billing_address>
        <first_name>John</first_name>
        <last_name>Doe</last_name>
        <address1>32, Doestreet</address1>
        <address2></address2>
        <zip_code>12345</zip_code>
        <city>New York</city>
      <state>NY</state>
      <country>US</country>
    </billing_address>
    <threeds>
      <authentication_flow>challenge</authentication_flow>
      <threeds_method>
        <status>completed</status>
      </threeds_method>
      <protocol>
        <target_version>2</target_version>
        <concrete_version>2</concrete_version>
      </protocol>
      <eci>05</eci>
    </threeds>
  </payment_transaction>
</wpf_payment>
```


Example Reconcile Response XML for successful payment with funds status:


```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <status>approved</status>
  <unique_id>26aa150ee68b1b2d6758a0e6c44fce4c</unique_id>
  <transaction_id>mtid20110408144716113553696 2</transaction_id>
  <consumer_id>123456</consumer_id>
  <timestamp>2011-04-08T14:46:27Z</timestamp>
  <amount>5000</amount>
  <currency>USD</currency>
  <payment_transaction>
    <status>approved</status>
    <transaction_type>ideal</transaction_type>
    <unique_id>bad08183a9ec545daf0f24c48361aa10</unique_id>
    <transaction_id>mtid20110408144716113553696 2</transaction_id>
    <terminal_token>e9fd7a957845450fb7ab9dccb498b6e1f6e1e3aa</terminal_token>
    <mode>test</mode>
    <timestamp>2011-04-08T14:46:40Z</timestamp>
    <descriptor>e-comprocessing.net/bogus +49123456789</descriptor>
    <funds_status>WAITING</funds_status>
    <account_holder>Name Surname</account_holder>
    <amount>4900</amount>
    <currency>GBP</currency>
    <customer_email>john.doe@example.com</customer_email>
    <customer_phone>+11234567890</customer_phone>
    <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
```

```
    <sca_exemption_result>13</sca_exemption_result>
    <technical_message>TESTMODE: No real money will be transferred!</technical_message>
    <message>TESTMODE: No real money will be transferred!</message>
  </payment_transaction>
</wpf_payment>
```

Example Response XML for voided payment:

```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <status>voided</status>
  <unique_id>26aa150ee68b1b2d6758a0e6c44fce4c</unique_id>
  <transaction_id>mtid20110408144716113553696 2</transaction_id>
  <consumer_id>123456</consumer_id>
  <timestamp>2011-04-08T14:46:27Z</timestamp>
  <amount>5000</amount>
  <currency>USD</currency>
  <usage>Shopify Electronic Transaction</usage>
  <description>You are about to buy shoes from Shopify</description>
  <card_brand>visa</card_brand>
  <card_number>420000...0000</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_holder>John Doe</card_holder>
  <expiration_year>2020</expiration_year>
  <expiration_month>2</expiration_month>
  <payment_transaction>
    <status>voided</status>
    <authorization_code>345678</authorization_code>
    <scheme_response_code>00</scheme_response_code>
    <response_code>00</response_code>
    <transaction_type>sale</transaction_type>
    <unique_id>bad08183a9ec545daf0f24c48361aa10</unique_id>
    <transaction_id>mtid20110408144716113553696 2</transaction_id>
    <arn>74537605259536043849425</arn>
    <terminal_token>e9fd7a957845450fb7ab9dccb498b6e1f6e1e3aa</terminal_token>
    <mode>test</mode>
    <timestamp>2011-04-08T14:46:40Z</timestamp>
    <descriptor>e-comprocessing.net/bogus +49123456789</descriptor>
    <bank_account_number>Bank Account Number</bank_account_number>
    <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
    <amount>5000</amount>
    <currency>USD</currency>
    <reason_for_not_honoring_exemption>8A01</reason_for_not_honoring_exemption>
    <sca_exemption_result>13</sca_exemption_result>
    <customer_email>john.doe@example.com</customer_email>
    <customer_phone>+11234567890</customer_phone>
    <billing_address>
     <first_name>John</first_name>
     <last_name>Doe</last_name>
     <address1>32, Doestreet</address1>
     <address2></address2>
     <zip_code>12345</zip_code>
     <city>New York</city>
     <state>NY</state>
     <country>US</country>
    </billing_address>
  </payment_transaction>
  <payment_transaction>
    <status>approved</status>
    <authorization_code>345678</authorization_code>
    <scheme_response_code>00</scheme_response_code>
    <response_code>00</response_code>
    <transaction_type>void</transaction_type>
    <unique_id>bad08183a9ec545daf0f24c48361aa10</unique_id>
    <transaction_id>mtid20110408144716113553696 2</transaction_id>
    <terminal_token>e9fd7a957845450fb7ab9dccb498b6e1f6e1e3aa</terminal_token>
    <mode>test</mode>
    <timestamp>2011-04-08T14:46:40Z</timestamp>
    <descriptor>e-comprocessing.net/bogus +49123456789</descriptor>
    <amount>5000</amount>
    <currency>USD</currency>
  </payment_transaction>
</wpf_payment>
```

Example Error Response XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <status>error</status>
  <code>100</code>
  <technical_message>Unknown system error. Please contact support.</technical_message>
  <message>Transaction failed, please contact support!</message>
</wpf_payment>
```

#### RECONCILE BY DATE RANGE

Reconcile by Date can be used to retrieve data about payments in date and time range. This can be useful if the merchant wants to retrieve information about payments belonging to the date and time range. For convenience, there are options to define records per page and the exact page of interest.

Reconcile by Date requests are handled exactly like transaction requests in the Processing API. To a large degree, the WPF Reconcile follows the notions of the standard Reconcile in Processing API.

The URL for date range reconcile is:

`https://staging.wpf.e-comprocessing.net/wpf/reconcile/by_date`

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf/reconcile/by_date \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
  <?xml version="1.0" encoding="UTF-8"?>
  <wpf_reconcile>
    <start_date>2024-11-29 16:01:50</start_date>
    <end_date>2025-02-27 16:01:50</end_date>
    <page>1</page>
    <records_per_page>100</records_per_page>
  </wpf_reconcile>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| start_date | required | yyyy-mm-dd hh:mm:ss | Start date and time of the reconcile range (time is optional) |
| end_date | optional | yyyy-mm-dd hh:mm:ss | End of date and time of the reconcile range (time is optional) |
| page | optional | string(11) | Number of the page |
| records_per_page | optional | 1 to 3 digits | Number ot records per page |

required* = conditionally required

Successful Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment_responses page="1" per_page="100" total_count="1" pages_count="1">
  <wpf_payment>
    <status>approved</status>
    <unique_id>e27bd9472d40d3734a49e3e5c5ada52d</unique_id>
    <transaction_id>246d15a8644ff203fca6ae39fac8e999</transaction_id>
    <timestamp>2025-02-07</timestamp>
    <amount>500</amount>
    <currency>USD</currency>
    <authorization_code>345678</authorization_code>
    <retrieval_reference_number>016813015184</retrieval_reference_number>
  </wpf_payment>
</wpf_payment_responses>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the WPF transaction, see states |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| reason_for_not_honoring_exemption | string | Reason for not honoring exemption. Check SCA Reason For Not Honoring Exemption Values. |
| sca_exemption_result | string | SCA exemption result. Check SCA Exemption Result Values. |

Error Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <status>error</status>
  <code>340</code>
  <technical_message>start_date has an invalid format</technical_message>
  <message>Please check input data for errors!</message>
</wpf_payment>
```

Error Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <status>error</status>
  <code>320</code>
  <technical_message>start_date is missing</technical_message>
  <message>Please check input data for errors!</message>
</wpf_payment>
```

Error Response

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <status>error</status>
  <code>340</code>
  <technical_message>per_page has an invalid format</technical_message>
  <message>Please check input data for errors!</message>
</wpf_payment>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the WPF transaction, see states |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

**WPF STATES**

WPF transactions will have one of the following states:

| Status | Description |
|---|---|
| new | Initial status of the wpf transaction. |
| user | Waiting for the user to fill out and submit the WPF form. |
| timeout | The transaction has been pending for too long and the timeout period has been reached. |
| in_progress | The transaction is currently being processed. |
| unsuccessful | Currently not used. |
| pending | The outcome of the transaction could not be determined, e.g. due to a timeout. The transaction state will eventually change - perform a reconcile after a certain time frame to obtain the correct status. |
| pending_hold | An asynchronous transaction has been finalized by the user but is waiting for a final update from the provider. |
| approved | The transaction has been approved by the schemes and it has been successful. |
| declined | The transaction has been declined by the schemes or risk management. |

| | |
|---|---|
| error | An error has occurred while negotiating with the schemes. |
| refunded | Once an approved transaction has been refunded the state changes to refunded. |
| voided | The transaction has been authorized, but cancelled at a later stage by the merchant. |
| chargebacked | Once an approved transaction has been chargebacked - the state changes to chargebacked. A chargeback occurs when the cardholder or the issuer rejects an accepted transaction for which funds have already been transferred. |
| chargeback_reversal | Once a chargebacked transaction has been charged, the state changes to chargeback_reversal. The chargeback has been cancelled. |
| represented | When the merchant submits evidence to prove that a chargeback is illegitimate, a chargebacked transaction changes its state to represented. The chargeback has been dismissed. |
| representment_reversed | When a representment initiated by the merchant is cancelled or rejected by the acquirer (the merchant's bank) due to an error or invalid reason for representment. The funds that were initially refunded to the merchant as a result of the re-presentment are reversed back to the issuer account (the cardholder's bank). |
| second_chargebacked | Once a chargeback_reversed/represented transaction is chargebacked the state changes to second chargebacked. |
| pending_review | The transaction is on hold, a manual review needs to be performed. |
| submitted | The WPF form was submitted by the user. |

**ERRORS**

WPF returns the same error codes/XML as the regular Server-to-Server API methods (e.g. sale). See the Error Section for more details.

---

## WPF transaction types

The web payment form supports the typical card-related transaction types such as authorizes, sales, and init recurring - with and without 3D (see below). If more than one card-based transaction type is passed, then if the cardholder decides to pay with a credit card, the first card-based transaction type with a valid configuration - terminals, MIDs, currencies, etc - is selected and used when cardholder decides to pay with a (credit) card.

Also Account Verification for cards is supported by the web payment form. This is a transaction type for card verifications without any financial impact on cardholder's account. If ‘account_verification’ is the only transaction type provided, the ‘amount’ and the ’currency’ parameters are **NOT** mandatory. If it is combined with other transaction types, they are still mandatory and required.

In addition, the WPF API supports a host of alternative payment methods (APMs), bank transfer payments, wallets, and more. All of the alternative transaction types are offered to the cardholder for payment, so the cardholder can choose between credit cards and alternatives defined by the merchant. Only credit cards or only a single alternative method can also work, there is no need to always offer credit cards for example.

Merchant custom attributes for a number of the transaction types are required. The WPF API supports those merchant custom attributes - they are submitted as child elements to the transaction type they belong to(see WPF API example request above).

The web payment form has the concept of a default payment method. If a given transaction type has the ‘default’ attribute set to ‘true’, then this transaction type will be pre-selected as default when the cardholder is redirected to the web payment form. If more than one transaction type has the ’default’ attribute, the first one with this attribute will be pre-selected. In the case there is only one transaction type requested (with or without the ’default’=’true’), this transaction type is automatically set as default by design.

Note that for each transaction type requested by a merchant in the WPF API, a valid configuration should exist - valid terminal, currency, MID, web payment form enabled for the merchant, etc. - otherwise a configuration error will be raised. If this happens, contact the IT support team to resolve/configure your desired transaction types correctly.

Example Request XML with custom attributes:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <transaction_id>wev238f328nc</transaction_id>
  <usage>Order ID 500, Shoes</usage>
  <description>You are about to buy 3 shoes at www.shoes.com!</description>
  <notification_url>https://example.com/notification</notification_url>
  <return_success_url>https://example.com/return_success</return_success_url>
  <return_failure_url>https://example.com/return_failure</return_failure_url>
  <return_cancel_url>https://example.com/return_cancel</return_cancel_url>
  <amount>5000</amount>
  <currency>USD</currency>
  <customer_email>john.doe@example.com</customer_email>
  <customer_phone>+11234567890</customer_phone>
  <card_holder>john doe</card_holder>
  <billing_address>
    <first_name>John</first_name>
    <last_name>Doe</last_name>
    <address1>23, Doestreet</address1>
    <zip_code>11923</zip_code>
    <city>New York City</city>
    <state>NY</state>
    <country>US</country>
  </billing_address>
  <transaction_types>
    <transaction_type>
      <name>sale</name>
      <recurring_type>initial</recurring_type>
      <bin>420000</bin>
      <tail>0000</tail>
      <expiration_date>2017-03</expiration_date>
      <fx_rate_id>123</fx_rate_id>
      <crypto>true</crypto>
    </transaction_type>
    <transaction_type>
      <name>sale3d</name>
      <recurring_type>initial</recurring_type>
      <default>true</default>
      <tail>1111</tail>
      <expiration_date>2016-03</expiration_date>
      <fx_rate_id>123</fx_rate_id>
      <crypto>true</crypto>
    </transaction_type>
    <transaction_type name="cashu"/>
    <transaction_type name="ezeewallet">
      <source_wallet_id>john.doe@e-comprocessing.net</source_wallet_id>
    </transaction_type>
    <transaction_type name="ppro">
        <payment_method>mybank</payment_method>
    </transaction_type>
    <transaction_type name="insta_debit_payin">
        <customer_account_id>123456</customer_account_id>
    </transaction_type>
    <transaction_type name="invoice">
      <payment_type>secure_invoice</payment_type>
      <customer_birthdate>1989-09-21</customer_birthdate>
      <customer_reference_number>123123</customer_reference_number>
    </transaction_type>
  </transaction_types>
  <recurring_category>subscription</recurring_category>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
</wpf_payment>
```

| Transaction Type | Custom Attribute | Required | Description |
|---|---|---|---|
| **authorize** | | | A standard authorization |

| | | | |
|---|---|---|---|
| | bin | no | Card’s first 6 digits |
| | tail | no | Card’s last 4 digits |
| | default | no | Configure as default or not |
| | expiration_date | no | Expiration month and year |
| | gaming | no | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| | crypto | no | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |
| | moto | no | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| | fx_rate_id | no | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **authorize3d** | | | A 3D-based authorization |
| | bin | no | Card’s first 6 digits |
| | tail | no | Card’s last 4 digits |
| | default | no | Configure as default or not |
| | expiration_date | no | Expiration month and year |
| | gaming | no | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| | crypto | no | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |
| | moto | no | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| | fx_rate_id | no | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **sale** | | | A standard sale |
| | bin | no | Card’s first 6 digits |
| | tail | no | Card’s last 4 digits |
| | default | no | Configure as default or not |
| | expiration_date | no | Expiration month and year |
| | gaming | no | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| | crypto | no | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |
| | moto | no | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| | fx_rate_id | no | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **sale3d** | | | A 3D-based sale |
| | bin | no | Card’s first 6 digits |
| | tail | no | Card’s last 4 digits |
| | default | no | Configure as default or not |
| | expiration_date | no | Expiration month and year |
| | gaming | no | Signifies whether a gaming transaction is performed. Gaming transactions usually use MCC 7995. Contact tech-support@e-comprocessing.com for more details |
| | crypto | no | Signifies whether a purchase of crypto-currency transaction is performed. Must be populated when purchasing crypto-currency with a VISA card. Must be populated when purchasing crypto-currency with a MASTER or INTL MAESTRO card and MCC is one of 6051, 6211. Contact tech-support@e-comprocessing.com for more details |
| | moto | no | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| | fx_rate_id | no | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| **init_recurring_sale** | | | A standard init recurring |
| | bin | no | Card’s first 6 digits |
| | tail | no | Card’s last 4 digits |
| | default | no | Configure as default or not |
| | expiration_date | no | Expiration month and year |
| | moto | no | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| | fx_rate_id | no | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| | managed_recurring | no | See Managed Recurring. Offers the option to automatically schedule recurring transactions. Contact tech-support@e-comprocessing.com for more details |
| **init_recurring_sale3d** | | | A 3D-based init recurring |
| | bin | no | Card’s first 6 digits |
| | tail | no | Card’s last 4 digits |
| | default | no | Configure as default or not |

| | | | |
|---|---|---|---|
| | expiration_date | no | Expiration month and year |
| | moto | no | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| | fx_rate_id | no | See Get rates for FX Service. Offers the option to use a specific FX rate to convert the transaction processing amount. Used FX rate should have the same source currency as the processing currency. Contact tech-support@e-comprocessing.com for more details |
| | managed_recurring | no | See Managed Recurring. Offers the option to automatically schedule recurring transactions. Contact tech-support@e-comprocessing.com for more details |
| **account_verification** | | | Card verification without any financial impact |
| | bin | no | Card’s first 6 digits |
| | tail | no | Card’s last 4 digits |
| | default | no | Configure as default or not |
| | expiration_date | no | Expiration month and year |
| | moto | no | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech-support@e-comprocessing.com for more details |
| **alipay** | | | Alipay is an oBeP-style alternative payment method. |
| usage | optional | Description of the transaction for later use. | |
| default | no | Configure as default or not | |
| **argencard** | | | Argencard is a cash payment method in Argentina. |
| default | no | Configure as default or not | |
| **aura** | | | Aura is a cash payment method in Brazil. |
| default | no | Configure as default or not | |
| **bancomer** | | | BBVA Bancomer is a cash payment method in Mexico. |
| default | no | Configure as default or not | |
| **boleto** | | | Boleto Bancario is a cash payment method in Brazil. |
| default | no | Configure as default or not | |
| **bcmc** | | | Bancontact is a local Belgian debit card scheme. |
| default | no | Configure as default or not | |
| **baloto** | | | Baloto is a cash payment option in Colombia. |
| default | no | Configure as default or not | |
| **banco_do_brasil** | | | Banco do Brasil offers online bank transfer payment service. |
| default | no | Configure as default or not | |
| **bitpay_sale** | | | Alternative payment methods supporting digital cryptocurrencies. |
| | default | no | Configure as default or not |
| **bitpay_payout** | | | BitPay Payout is a crypto currency payout method. |
| crypto_address | required | string(255) | Valid crypto address where the funds will be received |
| crypto_wallet_provider | required | string(255) | If crypto wallet provider is not in the table below, you must send ’other’ |
| default | no | Configure as default or not | |
| **bradesco** | | | Bradesco is a payment service in Brazil. |
| default | no | Configure as default or not | |
| **cashu** | | | CashU is a voucher payment method |
| | default | no | Configure as default or not |
| **container_store** | | | The container store transactions are made using gift cards. |
| card_number | required | string(19..21) | Gift card number |
| cvv | required | 5 to 8 digits | Verification code of the gift card, requirement is based on terminal configuration |
| default | no | Configure as default or not | |
| **cabal** | | | Cabal is a card payment method in Argentina. |
| default | no | Configure as default or not | |
| **cencosud** | | | Cencosud is a cash payment method in Argentina. |
| default | no | Configure as default or not | |
| **davivienda** | | | Davivienda is offering the Bill pay service which is a fast, easy and secure |
| default | no | Configure as default or not | |
| **ezeewallet** | | | eZeeWallet is a comprehensive digital wallet. |
| merchant_website | required | url | Website URL of the merchant initiating the wallet transfer |
| default | no | Configure as default or not | |
| **e_wallet** | | | eWallet transaction that handles different e-wallet providers |
| payment_type | required | string | eWallet provider name: Airtel Money / Amazon pay / Free Charge / Jio Money / Ola Money / Paytm / Payzapp / PhonePe |
| default | no | Configure as default or not | |
| **efecty** | | | Efecty is an offline cash payment voucher option in Colombia. |
| default | no | Configure as default or not | |
| **elo** | | | Elo is a cash payment method in Brazil. |
| default | no | Configure as default or not | |

| | | | |
|---|---|---|---|
| **eps** | | | EPS is a cash payment method in Austria. |
| default | no | Configure as default or not | |
| **fashioncheque** | | | Fashioncheque transactions are made using gift card |
| default | no | Configure as default or not | |
| **apple_pay** | | | Apple Pay is a mobile payment solution available on iOS devices with Touch ID / Face ID support and allows shoppers to purchase with credit and debit cards linked to their devices. |
| payment_subtype | yes | Payment subtype: **authorize** / **sale** | |
| recurring_type | no | Recurring type: **initial** | |
| default | no | Configure as default or not | |
| **google_pay** | | | Google Pay is an option to use credit or debit cards connected to a consumer's Google account |
| payment_subtype | yes | Payment subtype: **authorize** / **sale** | |
| recurring_type | no | Recurring type: **initial** | |
| default | no | Configure as default or not | |
| **invoice** | | | APMs Invoice Transaction |
| payment_type | yes | Payment type: **klarna** / **secure_invoice** | |
| customer_birthdate | yes | Customer birthdate. For example: **1989-09-22** | |
| customer_reference_number | yes | Customer number in merchant system | |
| default | no | Configure as default or not | |
| **itau** | | | Itau is a cash payment method in Brazil. |
| default | no | Configure as default or not | |
| **ideal** | | | iDeal is the most popular payment method in the Netherlands. |
| default | no | Configure as default or not | |
| **idebit_payin** | | | iDebit is alternative payment method |
| default | no | Configure as default or not | |
| **insta_debit_payin** | | | Debit is alternative payment method |
| default | no | Configure as default or not | |
| **intersolve** | | | Intersolve transactions iDeal is the most popular payment method in the Netherlands. |
| default | no | Configure as default or not | |
| **multibanco** | | | Multibanco allows Portugese shoppers to do payments through the Internet by using virtual credit cards. |
| default | no | Configure as default or not | |
| **my_bank** | | | MyBank is a payment method for Italy and Spain. |
| default | no | Configure as default or not | |
| **naranja** | | | Naranja is a cash payment method in Argentina. |
| default | no | Configure as default or not | |
| **nativa** | | | Nativa is a cash payment method in Argentina. |
| default | no | Configure as default or not | |
| **neosurf** | | | Neosurf is voucher payment method supported via IPG |
| default | no | Configure as default or not | |
| **neteller** | | | Neteller is wallet payment method |
| | default | no | Configure as default or not |
| **online_banking** | | | Online Banking is an OBeP payment method. |
| | default | no | Configure as default or not |
| | bank_codes | no | List of top level bank codes |
| **oxxo** | | | OXXO is the preferred payment method in Mexico. |
| default | no | Configure as default or not | |
| **paysafecard** | | | PaySafeCard is a voucher payment method |
| default | no | Configure as default or not | |
| **post_finance** | | | Online Banking ePayment method |
| default | no | Configure as default or not | |
| **ppro** | | | Supports payments with EPS, TeleIngreso, SafetyPay, Przelewy24, iDEAL, Mr. Cash and MyBank |
| default | no | Configure as default or not | |
| **poli** | | | Payment by bfvank account for customers with an Australian or New Zealand bank account. |
| default | no | Configure as default or not | |
| **p24** | | | Payment by bank account for customers with a Polish bank account. |
| default | no | Configure as default or not | |
| **pay_pal** | | | PayPal gives an option to use consumer's PayPal account payment methods. |
| payment_type | yes | Payment type: **authorize** / **sale** / **express** | |

| | | | |
|---|---|---|---|
| default | no | Configure as default or not | |
| **payu** | | | PayU is a Czech payment method. |
| default | no | Configure as default or not | |
| **post_finance** | | | Post Finance is an online Switzerland banking method |
| default | no | Configure as default or not | |
| **pago_facil** | | | Pago Facil is a payment service in Argentina |
| default | no | Configure as default or not | |
| **pse** | | | Pagos Seguros en Linea (PSE) is a payment service in Colombia |
| default | no | Configure as default or not | |
| **upi** | | | UPI (Unified Payment Interface) transaction is an alternative payment method. |
| default | no | Configure as default or not | |
| **rapi_pago** | | | Rapipago is an Argentinian payment method used for online purchases. |
| default | no | Configure as default or not | |
| **redpagos** | | | Redpagos is a cash payment method in Uruguay. |
| default | no | Configure as default or not | |
| **santander** | | | Santander is an online bank transfer for ecommerce purchases. |
| **safetypay** | | | Safetypay is a real-time bank transfer system that operates in more than 10 different countries. |
| default | no | Configure as default or not | |
| **sofort** | | | Bank transfer payment, popular in Germany |
| default | no | Configure as default or not | |
| **webmoney** | | | Bank transfer payment, popular in Russian Federation. |
| **sdd_sale** | | | SEPA Direct Debit sale |
| **sdd_init_recurring_sale** | | | SEPA Direct Debit init recurring |
| **tarjeta_shopping** | | | Tarjeta Shopping is a cash payment method in Argentina. |
| default | no | Configure as default or not | |
| **trustly_sale** | | | Solution for Online Banking ePayments. |
| return_success_url_target | required | URLTarget for successful payment in Trustly iFrame. Possible values: **top**, **self**, **parent** | |
| **webpay** | | | Webpay is a payment solution in Chile that allows shoppers to pay online with their credit card. |
| default | no | Configure as default or not | |
| **wechat** | | | Online Banking ePayment method |
| | product_code | no | Product code |
| | product_num | no | Product number |
| | product_desc | no | Product description |
| **invoice** | | | APMs Invoice Transaction |
| | payment_type | yes | Payment type: **klarna** / **secure_invoice** |
| | customer_birthdate | yes | Customer birthdate. For example: **1989-09-22** |
| | customer_reference_number | yes | Customer number in merchant system |
| **russian_mobile_sale** | | | Mobile network operator payments in Russian Federation. |
| | target | required | Payment target |
| | operator | required | Mobile network operator name (**mtc**, **megafon**, **tele2** or **beeline**). |

---

## WPF Internationalization (i18n)

The web payment form is internationalized (i18n) and supports the following locales and corresponding languages:

| Locale | Language | Description |
|---|---|---|
| en | English | English locale and language settings (this is the default) |
| it | Italian | Italian locale and language settings |
| es | Spanish | Spanish locale and language settings |
| fr | French | French locale and language settings |
| de | German | German locale and language |
| pl | Polish | Polish locale and language |
| ja | Japanese | Japanese locale and language |
| zh | Mandarin Chinese | Mandarin Chinese locale and language |
| ar | Arabic | Arabic locale and language |
| pt | Portuguese | Portuguese locale and language |
| tr | Turkish | Turkish locale and language |
| ru | Russian | Russian locale and language |
| hi | Hindu | Hindu locale and language |

| | | |
|---|---|---|
| bg | Bulgarian | Bulgarian locale and language |
| nl | Dutch | Dutch locale and language |
| is | Icelandic | Icelandic locale and language |
| id | Indonesian | Indonesian locale and language |
| ms | Malay | Malay locale and language |
| th | Thai | Thai locale and language |
| cs | Czech | Czech locale and language |
| hr | Croatian | Croatian locale and language |
| sl | Slovenian | Slovenian locale and language |
| fi | Finnish | Finnish locale and language |
| no | Norwegian | Norwegian locale and language |
| da | Danish | Danish locale and language |
| sv | Swedish | Swedish locale and language |

Note that the English locale is the default, and if you don't specify another locale in the WPF API, this is the locale and language that the web payment form will be translated into.

The customer has the option to change the language once he has been redirected to the WPF via the received `redirect_url`.

It is a good practice to submit the desired locale in the WPF API create call, so that a proper `redirect_url` is returned instead of manually parsing and generating a locale-specific `redirect_url`.

If a locale/language different than the current ones is needed or any translation errors/inconsistencies are spotted, feel free to contact the IT Support team at tech-support@e-comprocessing.com and contribute to the translation effort.

---

# 3DSecure

---

3DSecure (3-domain structure), also known as a payer authentication, is a security protocol that helps to prevent fraud in online credit and debit card transactions. This additional security was initiated and created by **Visa** and **MasterCard** and it's branded as **Verified by Visa** and **MasterCard SecureCode** respectively.

**3DSecure (3DS)**: A set of protocols and procedures through which a customer's ownership of the credit card they are using can be assessed. 3DSv2 offers a number of enhancements, including frictionless flows.

---

## V2

#### INTRODUCTION

How does 3DS 2.0 work? Better, stronger fraud-detection intelligence, to put it simply.

3DSecure has been around for years and creates an authentication data connection between digital merchants, payment networks and financial institutions to be able to analyze and share more intelligence about transactions. The new 2.0 version of the technology enables a real-time, secure, information-sharing pipeline that merchants can use to send an unprecedented number of transaction attributes that the issuer can use to authenticate customers more accurately without asking for a static password or slowing down commerce.

All in all, the second iteration of the 3DSecure protocol is a great improvement for all parties involved.

It allows merchants to provide protection across multiple platforms with easy integration into their systems, including mobile applications, while still being able to exploit the benefits that the protocol provides. It is also estimated that cart abandonment rates will dramatically fall.

Issuers can share and receive more data with merchants, giving them a greater insight into transactional patterns which will allow them to determine the risk with higher accuracy and therefore improve the authentication process. 3DS 2.0 will also offer banks the opportunity to effortlessly comply with the requirements of PSD2.

For customers, the updates are perhaps most beneficial. They can now enjoy protection from fraudulent transactions across most platforms.

Not only will transactions be more secure through the implementation of increased protection methods such as two-factor authentication (2FA), but the consumer experience will also be greatly improved by frictionless flows through risk-based authentication.

3DS 2.0 is an updated protocol created to recognise the need of current and future requirements including adding the support of mobile-based authentication and digital wallets integration. One of the main benefits is the introduction of a frictionless flow that allows issuers to approve a transaction without requiring any manual input from the cardholder.

**3DSecure Challenge**: The 3DS 2.0 step during which the customer interacts with the card issuer to provide additional authentication for the transaction.

**3DSecure Method**: Refers to the 3DS 2.0 step during which the card issuer gathers information about the customer via their browser. A card issuer may use this data to help make a risk-based decision , for example if the issuer can identify that the cardholder has used this web browser to make purchases in the past, they may decide the transaction is low risk and consider the customer authenticated without requiring a 3DSecure Challenge.

**3DSecure Method Handling**: The 3DSecure Method allows for additional browser information to be gathered by an ACS prior to receipt of the AReq message to help facilitate the transaction risk assessment. The use of the 3DSecure Method by an ACS is optional. The inclusion of 3DSecure Method URL and account ranges in a Directory Server is optional for an ACS.

**Access Control Server**: The ACS contains the authentication rules and is controlled by the Issuer. ACS functions include:

- Verifying whether a card number is eligible for 3DSecure authentication
- Verifying whether a Consumer Device type is eligible for 3DSecure authentication
- Authenticating the Cardholder for a specific transaction

**Consumer Device**: The Consumer Device has the capability to run a 3DSecure Requestor App or present a website on a browser that can be used for 3DSecure authentication. The Consumer Device-based components of the 3DSecure Requestor Environment depend on the model:

- App-based - the 3DSecure SDK integrated with the 3DSecure Requestor App
- Browser-based - a browser utilising the 3DSecure Method

**Frictionless**: The authentication completes without requiring a interaction between consumer and issuer.

#### REQUEST PARAMS

> In order to enforce the 3DSecure v2 authentication protocol, the request params below must be submitted in the transaction request for: Authorize3d Sale3d InitRecurringSale3d.

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **threeds_v2_params** | required* | | 3DSv2 async parameters. They must be submitted in order to use the 3DSv2 authentication protocol in asynchronous workflow |
| **threeds_method** | optional | | 3DS-Method related parameters for any callbacks and notifications. |
| callback_url | optional | url | Specific 3DS-Method callback URL after the 3DS-Method completes. The actual status will be provided via HTTP POST to that URL. For more information, go to 3DSv2 method params |
| **control** | required* | | General params for preferences in authentication flow and providing device interface information. |
| device_type | required* | string | Identifies the device channel of the consumer, **required** in the 3DSv2 authentication protocol. For more information, go to 3DSv2 control params |
| challenge_window_size | required* | string | Identifies the size of the challenge window for the consumer. For more information, go to 3DSv2 control params |

| | | | |
|---|---|---|---|
| challenge_indicator | optional | string | The value has weight and might impact the decision whether a challenge will be required for the transaction or not. If not provided, it will be interpreted as **no_preference**. For more information, go to 3DSv2 control params |
| **purchase** | optional | | Purchase related params providing with additional information regarding the order. |
| category | optional | string | Identifies the type of transaction being authenticated. This field is required in some markets.Accepted values are: **goods**, **service**, **check_acceptance**, **account_funding**, **quasi_cash**, **prepaid_activation**, **loan**. |
| **merchant_risk** | recommended | | Merchant risk assessment params. They are all optional, but recommended. |
| shipping_indicator | optional | string(16) | Indicator code that most accurately describes the shipping method for the cardholder specific transaction. If one or more items are included in the sale, use the Shipping Indicator code for the physical goods. If all digital goods, use the code that describes the most expensive item. Accepted values are: **same_as_billing**, **stored_address**, **verified_address**, **pick_up**, **digital_goods**, **travel**, **event_tickets**, **other**. |
| delivery_timeframe | optional | string(11) | Indicates the merchandise delivery timeframe. Accepted values are: **electronic**, **same_day**, **over_night**, **another_day**. |
| reorder_items_indicator | optional | string(10) | Indicates whether the cardholder is reordering previously purchased merchandise. Accepted values are: **first_time**, **reordered**. |
| pre_order_purchase_indicator | optional | string(21) | Indicates whether cardholder is placing an order for merchandise with a future-availability or release date. Accepted values are: **merchandise_available**, **future_availability**. |
| pre_order_date | optional | dd-mm-yyyy | For a pre-ordered purchase, the expected date that the merchandise will be available. |
| gift_card | optional | 'true' | Prepaid or gift card purchase. |
| gift_card_count | optional | integer | For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased. The value is limited to **99**. |
| **card_holder_account** | recommended | | Cardholder account additional information. They are all optional, but recommended, because they have a significant impact on approval rates |
| creation_date | optional | dd-mm-yyyy | Date that the cardholder opened the account with the 3DS Requester. |
| update_indicator | optional | string(19) | Length of time since the cardholder’s account information with the 3DS Requestor was last changed. Includes Billing or Shipping address, new payment account, or new user(s) added. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| last_change_date | optional | dd-mm-yyyy | Date that the cardholder’s account with the 3DS Requestor was last changed. Including Billing or Shipping address, new payment accout, or new user(s) added. |
| password_change_indicator | optional | string(18) | Length of time since the cardholder account with the 3DS Requestor had a password change or account reset. Accepted values are: **no_change**, **during_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| password_change_date | optional | dd-mm-yyyy | Date that cardholder’s account with the 3DS Requestor had a password change or account reset. |
| shipping_address_usage_indicator | optional | string(19) | Indicates when the shipping address used for this transaction was first used with the 3DS Requestor. Accepted values are: **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| shipping_address_date_first_used | optional | dd-mm-yyyy | Date when the shipping address used for this transaction was first used with the 3DS Requestor. |
| transactions_activity_last_24_hours | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours. |
| transactions_activity_previous_year | optional | integer | Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year. |
| provision_attempts_last_24_hours | optional | integer | Number of Add Card attempts in the last 24 hours. |
| purchases_count_last_6_months | optional | integer | Number of purchases with this cardholder account during the previous six months. |
| suspicious_activity_indicator | optional | string(22) | Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account. Accepted values are: **no_suspicious_observed**, **suspicious_observed**. |
| registration_indicator | optional | string(19) | Indicates the length of time that the payment account was enrolled in the cardholder’s account with the 3DS Requester. Accepted values are: **guest_checkout**, **current_transaction**, **less_than_30days**, **30_to_60_days**, **more_than_60days**. |
| registration_date | optional | dd-mm-yyyy | Date that the payment account was enrolled in the cardholder’s account with the 3DS Requestor. |
| **browser** | required* | | For browser-based transactions. They are all *required* in case the **device_type** is set to **browser** |
| accept_header | required* | string(2048) | The exact content of the HTTP **ACCEPT** header as sent to the 3DS Requester from the Cardholder browser. Any other header different than the **ACCEPT** header will be rejected. Example: `application/json,text/plain,text/html,*/*`. |
| java_enabled | required* | boolean | Boolean that represents the ability of the cardholder browser to execute Java. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.javaEnabled`. |
| language | required* | string(8) | Value representing the browser language as defined in IETF BCP47. Note that only one browser language tag is about to be submitted as per the above **IETF BCP47**. Numeric chars are also allowed in the subtag and will represent the region.Example: `en-GB`, `zh-guoyu`, `fil-PH`, `gsw`, `es-419`, `de-1996`, etc. The value can be retrieved by accessing a property of the navigator with JavaScript, `navigator.language`. |
| color_depth | required* | integer | Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from Cardholder browser using the `screen.colorDepth` property. The value as per EMVCo specs can be one of **1**, **4**, **8**, **15**, **16**, **24**, **32**, **48**. In case, an unsupported `color_depth` is determined, the nearest supported value that is less than the actual one needs to be submitted. For example, if the obtained value is **30**, which is not supported as per EMVCo specs, **24** has to be submitted. |
| screen_height | required* | integer | Total height of the Cardholder's screen in pixels. Value is returned from the screen.height property. |
| screen_width | required* | integer | Total width of the Cardholder's screen in pixels. Value is returned from the screen.width property. |
| time_zone_offset | required* | string(5) | Time difference between UTC time and the Cardholder browser local time, in **minutes**. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. If **UTC -5** hours then submit `300` or `+300`, If **UTC +2** hours then `-120`. The value can be retrieved using Javascript `getTimezoneOffset()` method over **Date** object. |
| user_agent | required* | string(2048) | Exact content of the HTTP user-agent header. |
| **sdk** | required* | | For application-based transactions. They are all *required* in case the **device_type** is set to **application** |
| interface | required* | string(6) | SDK Interface types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. Accepted values are: **native**, **html**, **both**. |
| **ui_types** | required* | | Lists all UI types that the device of the consumer supports for displaying specific challenge interfaces within the SDK. |
| ui_type | required* | string(13) | UI type that the device of the consumer supports for displaying specific challenge interface. Accepted values are: **text**, **single_select**, **multi_select**, **out_of_bag**, **other_html**. |
| application_id | required* | string(36) | Universally unique ID created upon all installations and updates of the 3DS Requestor APp on a Customer Device. This will be newly generated and stored by the 3DS SDK for each installation or update. The field is limited to 36 characters and it shall have a canonical format as defined in IETF RFC 4122. |
| encrypted_data | required* | string(64000) | JWE Object as defined Section 6.2.2.1 containing data encrypted by the SDK for the DS to decrypt. The data will be present when sending to DS, but not present from DS to ACS. |
| ephemeral_public_key_pair | required* | string(256) | Public key component of the ephemeral key pair generated by the 3DS SDK and used to establish session keys between the 3DS SDK and ACS. In AReq, this data element is contained within the ACS Signed Content JWS Object. The field is limited to maximum 256 characters. |
| max_timeout | required* | integer | Indicates the maximum amount of time (in minutes) for all exchanges. The field shall have value greater or equals than 05. |
| reference_number | required* | string(32) | Identifies the vendor and version of the 3DS SDK that is integrated in a 3DS Requestor App, assigned by EMVCo when the 3DS SDK is approved. The field is limited to 32 characters. |
| **recurring** | optional | | Additional recurring details. |
| expiration_date | optional | dd-mm-yyyy | A future date indicating the end date for any further subsequent transactions. For more information, go to 3DSv2 recurring params |

| | | | |
|---|---|---|---|
| frequency | optional | integer | Indicates the minimum number of days between subsequent transactions. An empty value indicates the payment frequency is not set.For more information, go to 3DSv2 recurring params |

`required* = conditionally required`

**3DS-METHOD PARAMS**

`callback_url` - *optional*

> Please, make sure the **callback_url** can accept HTTP POST request and the 3DS-Method status is handled properly.

**Notes:** The `callback_url` is used in 3DSv2 scope is case a 3DS-Method is required to be submitted in asynchronous workflow. The 3DS-Method is about to be submitted in an iframe and the final HTTP POST request to the `callback_url` will be initiated inside the iframe target of the HTML form used for submitting the 3DS-Method.

Once the 3DS-Method reaches a final state, a callback will be sent via HTTP POST (**application/x-www-form-urlencoded**) with the following parameters:

```
unique_id=44177a21403427eb96664a6d7e5d5d48
&threeds_method_status=completed
&signature=2e24755dc8622ecf14023852a669fb69a3d61ec2056e60688b8bddf29347f77362b1229c2a98d4a9339e1f685bcb90bd9b0638c025161744f30040db07452c3a
```

The signature is a **SHA512** of the concatenated string: `unique_id`, `threeds_method_status` and the `password` (**53ef627e5f9ac4766a8b872a2e0a8548370f0cd2**) used for the HTTP Basic authentication to the API.

For more detailed information about the 3DS-Method submission and the usage of the 3DS-Method `callback_url`, please take a look at the 3DSv2 authentication flow diagrams related to the asynchronous 3DS-Method submission.

**CONTROL PARAMS**

`challenge_indicator` - *optional*

| Value | Description |
|---|---|
| **no_preference** *(default)* | Don't have any preferences related to the Challenge flow |
| **no_challenge_requested** | I prefer that a Challenge flow does not take place |
| **preference** | A request for the Challenge flow to take place |
| **mandate** | A Challenge flow must take place to fulfill a mandate |

`device_type` - *required*

| Value | Description |
|---|---|
| **browser** | Browser-based consumer interface. All the `browser` parameters are required. |
| **application** | Application-based consumer interface, mobile etc. All the `sdk` parameters are required. |

`challenge_window_size` - *required**

> The **challenge_window_size** is required when the **device_type** is set to **browser**

**Notes:** *Dimensions of the challenge window that has been displayed to the Cardholder. The ACS shall reply with content that is formatted to appropriately render in this window to provide the best possible consumer experience. Preconfigured sizes are width X height in pixels of the window displayed in the Cardholder browser window.*

| Value | Description |
|---|---|
| **250x400** | Challenge window: width of `250px` and height of `400px` |
| **390x400** | Challenge window: width of `390px` and height of `400px` |
| **500x600** | Challenge window: width of `500px` and height of `600px` |
| **600x400** | Challenge window: width of `600px` and height of `400px` |
| **full_screen** | Challenge in a full screen |

**RECURRING PARAMS**

> All recurring params below are optional for asynchronous InitRecurringSale3d that is about to use the 3DSv2 authentication protocol.

> All recurring params below are optional for the Web Payment Form that has the InitRecurringSale3d in **transaction_types** and is about to use the 3DSv2 authentication protocol.

`expiration_date` - *optional*

> As a best practice, merchants should have a recurring expiry associated with all recurring transactions, but in cases like subscriptions where there is no established expiry or end date of recurring transactions, the param should be omitted.

| Value | Description |
|---|---|
| *27-08-2025* | No further subsequent transactions are expected after that date. |

`frequency` - *optional*

> The **frequency** can be a positive number, not lower than **1** and not greater than **9999**. An empty value indicates the payment frequency is not set.

| Value | Description |
|---|---|
| **14** | Indicates a minimum number of 14 days between subsequent recurring transactions. |
| **30** | Indicates a minimum number of 30 days *(monthly subscription)* between subsequent recurring transactions. |

**AUTHENTICATION FLOWS**

There are two general workflows for 3DS 2.0, **Frictionless** Flow and **Challenge** Flow.

> Transactions can fail prior to invoking the 3D mechanism (in case of e.g. invalid card number, expiry date or risk checks). In this case, the 3D transaction becomes synchronous.

> If the card is not participating in 3DS and non-participating cards are allowed on the MID configuration, the async 3D transaction behaves like synchronous and proceeds with the authorization message.

Please, use only HTTP **GET** method in order to navigate consumer to any of the redirect URLs in the 3DS authentication scope. The other HTTP methods: **POST**, **PUT**, etc are not supported for redirect URLs and won't be accepted for redirection. HTTP **POST** and **PUT** are accepted in other scenarios for the 3DS-Method.

**FRICTIONLESS**

A Frictionless Flow occurs when the Issuer authenticates the cardholder without cardholder involvement by evaluating the transaction's risk level.

The customer has more frictionless experience through the merchant’s platform by not being challenged. That means the drop-off rate due to the 3DSecure protocol will be drastically reduced and the customer will happily come back to the merchant’s platform.

The Frictionless Flow does not require further Cardholder interaction to achieve a successful authentication and complete the 3DSecure authentication process. The payment will be completely synchronous, the consumer won't be redirected to the Access Control Server to complete the authentication.

In order to enforce using the 3DSv2 authentication protocol, make sure to **include** the **threeds_v2_params** in the transaction request.

1.) The merchant submit a 3D transaction to the API including the 3DSecure v2 params

2.) No 3DSecure Method or 3DSecure Challenge is required by the card issuer, the transaction will be completed without further action.

3.) A synchronous payment response is returned to the merchant indicating the status of the payment.

An exemption from Strong Customer Authentication (SCA) can be requested by submitting an **exemption** with `low_risk` under SCA params.

In case the issuer accepts the exemption, a step up in the authentication flow might not be required because the transaction's risk analysis has already been performed by acquirer.

Note, the requested exemption might not be accepted due to internal risk validations.

For example, to be able to utilize the low risk exemption, the BIN country of the card must be part of the European Economic Area (EEA).

Furthermore, the acquirer could accept the merchant low-risk exemption request only if the transaction amount does not exceed the acquirer low-risk exemption threshold.

Finally, the ACS might not acknowledge the merchant/acquirer's exemption request and may still require a step up in the cardholder authentication.

Asynchronous 3 D Sv2 Frictionless Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4012000000060085</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <threeds_v2_params>
     <threeds_method>
       <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
     </threeds_method>
     <control>
       <device_type>browser</device_type>
       <challenge_window_size>full_screen</challenge_window_size>
       <challenge_indicator>preference</challenge_indicator>
     </control>
     <purchase>
       <category>service</category>
     </purchase>
     <merchant_risk>
       <shipping_indicator>verified_address</shipping_indicator>
       <delivery_timeframe>electronic</delivery_timeframe>
       <reorder_items_indicator>reordered</reorder_items_indicator>
       <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
       <pre_order_date>27-03-2025</pre_order_date>
       <gift_card>true</gift_card>
       <gift_card_count>2</gift_card_count>
     </merchant_risk>
     <card_holder_account>
       <creation_date>27-02-2024</creation_date>
       <update_indicator>more_than_60days</update_indicator>
       <last_change_date>27-11-2024</last_change_date>
       <password_change_indicator>no_change</password_change_indicator>
       <password_change_date>12-02-2025</password_change_date>
       <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
       <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
       <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
       <transactions_activity_previous_year>10</transactions_activity_previous_year>
       <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
       <purchases_count_last_6_months>5</purchases_count_last_6_months>
       <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
       <registration_indicator>30_to_60_days</registration_indicator>
       <registration_date>27-02-2023</registration_date>
     </card_holder_account>
     <browser>
       <accept_header>*/*</accept_header>
       <java_enabled>false</java_enabled>
       <language>en-GB</language>
       <color_depth>24</color_depth>
       <screen_height>900</screen_height>
       <screen_width>1440</screen_width>
       <time_zone_offset>-120</time_zone_offset>
       <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
     </browser>
     <sdk>
       <interface>native</interface>
       <ui_types>
         <ui_type>multi_select</ui_type>
       </ui_types>
```

```
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```

Frictionless Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>approved</status>
  <mode>test</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <threeds>
    <eci>05</eci>
  </threeds>
</payment_response>
```

Frictionless Declined Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>sale3d</transaction_type>
  <status>declined</status>
  <cvv_result_code>P</cvv_result_code>
  <retrieval_reference_number>323623238231</retrieval_reference_number>
  <scheme_response_code>06</scheme_response_code>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <consumer_id>123456</consumer_id>
  <response_code>05</response_code>
  <code>500</code>
  <technical_message>Do not honour</technical_message>
  <message>Transaction declined, please contact support!</message>
  <mode>test</mode>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <threeds>
    <eci>05</eci>
  </threeds>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
</payment_response>
```

**DIAGRAM**

**RECONCILE**

Once the transaction reaches the final state, a single reconcile can also be performed to retrieve more detailed information about the 3D transaction. It should include information about the 3DS transaction as described in the reconcile request/response below, on the right.

Reconcile 3 D Transaction By Unique Id Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
 </reconcile>'
```

Successful Reconciliation Of Frictionless 3 D Sv2 Transaction Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>sale3d</transaction_type>
   <status>approved</status>
   <authorization_code>005645</authorization_code>
   <retrieval_reference_number>016813015184</retrieval_reference_number>
   <payment_account_reference>Q1HJZ8RKA1EBL470G9XYG90R5D3E</payment_account_reference>
   <response_code>00</response_code>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <mode>test</mode>
   <timestamp>2025-02-26T16:01:50Z</timestamp>
   <descriptor>Descriptor one</descriptor>
```

```
    <amount>100</amount>
    <currency>USD</currency>
    <card_brand>visa</card_brand>
    <card_number>401200...0085</card_number>
    <card_type>CREDIT</card_type>
    <card_subtype>CARD SUBTYPE</card_subtype>
    <card_issuing_bank>Issuing Bank</card_issuing_bank>
    <card_issuing_country>Exact Issuing country</card_issuing_country>
    <bank_account_number>Bank Account Number</bank_account_number>
    <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
    <sent_to_acquirer>true</sent_to_acquirer>
    <arn>74537605259536043849425</arn>
    <scheme_response_code>00</scheme_response_code>
    <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
    <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
    <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
    <threeds>
      <authentication_flow>frictionless</authentication_flow>
      <protocol>
        <target_version>2</target_version>
        <concrete_version>2</concrete_version>
        <sub_version>2</sub_version>
      </protocol>
      <eci>05</eci>
    </threeds>
  </payment_response>
```


**Successful Reconcile Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ’true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| **threeds** | | |
| authentication_flow | string(255) | Identifies the concrete authentication flow of the 3DS transaction that it has gone through. It will be included only if the transaction reaches the final state. The possible values for 3DSv2 are **frictionless**, **challenge**. |
| **protocol** | | |
| target_version | integer | Identifies the requested version of the 3DS authentication protocol to be used. The possible values are **2**. |
| concrete_version | integer | Identifies the concrete version of the 3DS authentication protocol that the transaction has been processed through. The possible values are **2**. |
| eci | string(2) | See Electronic Commerce Indicator for details |

**NOTIFICATION**

Once the transaction reaches final state, a notification will be sent to the `notification_url` submitted in the initial transaction request. For more information, go to Asynchronous Transactions and Notifications.

Approved Notification Example for frictionless flow


```
?transaction_id=119643250547501c79d8295
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=sale3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=approved
&amount=100
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&eci=05
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&threeds_authentication_flow=frictionless
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```


Declined Notification Example for frictionless flow


```
?transaction_id=119643250547501c79d8295
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=sale3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=declined
&amount=100
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&eci=05
```

```
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&threeds_authentication_flow=frictionless
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

**3DS Attributes**

| Name | Type | Description |
|---|---|---|
| threeds_authentication_flow | string | Identifies the concrete 3DS authentication flow that the transaction has gone through. It will be available in the notification only if the consumer has finished the 3DS authentication with the issuer. The available values for 3DSv2 are frictionless and challenge. |
| threeds_method_status | string | Identifies the status of the 3DS-Method in the scope of 3DSv2 authentication protocol. The possible values are **required**, **in_progress** and **completed**. |
| threeds_target_protocol_version | string(1) | Identifies the 3DS protocol that has been enforced. The possible values are **2**. |
| threeds_concrete_protocol_version | string(1) | Identifies the concrete 3DS protocol version that the transaction has gone through. The possible values are **2**. |
| threeds_protocol_sub_version | string(1) | Identifies the 3DS protocol sub-version that the transaction has gone through. The possible values are **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**. |
| threeds_authentication_status_reason_code | string(2) | See Status Reason Code for details. |

Please have in mind, the above 3DS related params will be available for all transactions supporting 3DS in async workflow: Authorize3d, Sale3d, or InitRecurringSale3d. For more information about the 3DS transactions, go to 3DS Card.

#### FRICTIONLESS WITH 3DSECURE METHOD

A Frictionless Flow occurs when the Issuer authenticates the cardholder without cardholder involvement by evaluating the transaction's risk level.

The customer has more frictionless experience through the merchant’s platform by not being challenged. That means the drop-off rate due to the 3DSecure protocol will be drastically reduced and the customer will happily come back to the merchant’s platform.

The Frictionless Flow does not require further Cardholder interaction to achieve a successful authentication and complete the 3DSecure authentication process. The payment will be completely synchronous, the consumer won't be redirected to the Access Control Server to complete the authentication.

In order to enforce using the 3DSv2 authentication protocol, make sure to **include** the **threeds_v2_params** in the transaction request.

This flow is treated frictionless (without consumer interaction with the issuer), but with the only difference that the ACS requires a 3DSecure Method to be submitted before continuing.

Thanks to risk-based authentication performed in the ACS, frictionless flow allows issuers to approve a transaction without the need to interact with the consumer.

When the customer makes an online purchase they would add an item to their shopping cart, fill out the normal purchase information and then proceed to confirm the purchase.

Details of the purchase including device data, item purchased and value are submitted to the ACS server to determine the authenticity of the cardholder.

The ACS will then screen it with the risk-based elements. If the risk is deemed to be low, the ACS can authenticate the customer passively and not bother them with the extra confirmation.

This is a frictionless process for the customer as it happens behind the scenes. They are directed straight to the purchase confirmation screen, without even knowing that their transaction was screened.

An exemption from Strong Customer Authentication (SCA) can be requested by submitting an **exemption** with `low_risk` under SCA params.

In case the issuer accepts the exemption, a step up in the authentication flow might not be required because the transaction's risk analysis has already been performed by acquirer.

Note, the requested exemption might not be accepted due to internal risk validations.

For example, to be able to utilize the low risk exemption, the BIN country of the card must be part of the European Economic Area (EEA).

Furthermore, the acquirer could accept the merchant low-risk exemption request only if the transaction amount does not exceed the acquirer low-risk exemption threshold.

Finally, the ACS might not acknowledge the merchant/acquirer's exemption request and may still require a step up in the cardholder authentication.

**Asynchronous 3DS Method submission**

A link between the customer's browser and the card issuer must be opened with a hidden iframe. It is used for the card issuer to load JavaScript which gathers device information to be returned to the card issuer. The next step after initiating the iframe, is to submit an API call to the `threeds_method_continue_url` using **HTTP PUT**, to retrieve the next step in the authentication. The API request won't require HTTP basic authentication, but a proper **signature** needs to be included to prove the authenticity of the request. The response of this API call will be the same as the normal 3D transaction processing API response, but with the only difference that an additional interaction with the issuer might be requested.

The asynchronous submission of the 3DS-Method might look difficult to achieve, but mitigates the risk of potential transaction processing interruptions, because the consumer redirection does not depend on a successful 3DS-Method submission. A continuation of the 3DS-Method might be requested regardless of the 3DS-Method submission result. It's a responsibility of the ACS then to take the appropriate decision how to continue with the authentication.

In order to simulate this authentication flow, use a test card 4066330000000004 for frictionless flow that requires 3DS-Method and submit a 3DSv2 transaction in asynchronous workflow by including the `threeds_v2_params`.

The response of the API will indicate that further action is required:

- `status` - **pending_async**
- `threeds_method_url` - the URL action where the 3DS-Method needs to be submitted using HTTP POST
- `threeds_method_continue_url` - API endpoint that accepts HTTP PUT requests with a signature and returns transaction API response identifying what the next step is *(transaction completed or consumer interaction is needed)*

In order to submit a 3DS-Method, you need to create a hidden iframe in the consumer browser (client side) with an `html` and `body` tags as described below, on the right side and create a hidden **HTML form** that:

- targets the iframe
- uses HTTP METHOD **POST** - `method="post"`
- has an **action** equivalent to the value of `threeds_method_url`, received from the response of the initial transaction request
- has 2 hidden inputs
  - `unique_id` - equivalent to the value of the `unique_id`, received from the response of the initial transaction request
  - `signature` - **SHA512** of a concatenated string (**unique_id**, **amount**, **timestamp**, **merchant_api_password**), where **unique_id**, **amount**, **timestamp** can be taken from the response of the initial transaction request and **merchant_api_password** is the password used for HTTP Basic authentication to the API during the initial transaction request
- submit the HTML form in the background using JavaScript

Once the 3DS-Method submission is initiated, a callback via HTTP POST will be done inside the iframe when the 3DS-Method reaches the final state. The 3DS-Method callback will be sent to the `callback_url` (submitted in the 3DSv2 request params as described in the diagram below) and will be constructed as described below:

- with request headers
  - `Content-Type` - **application/x-www-form-urlencoded**
- with POST request params
  - `unique_id` - the exact **unique_id** of the transaction in the initial transaction request
  - `threeds_method_status` - the status of the 3DS-Method submission, expect a value of **completed**
  - `signature` - **SHA512** of a concatenated string with the values of **unique_id**, **threeds_method_status**, and **merchant_api_password** where **unique_id** and **threeds_method_status** are POST params and **merchant_api_password** is the password used for HTTP HTTP Basic authentication to the API during the initial transaction request

The callback above can be handled *(optional)* in order to get informed of the 3DS-Method status, whether it has completed or not.

The 3DS-Method callback is sent asynchronously as shown in the diagram.

In case you have implemented the 3DS-Method callback handler (it's not required), make sure you validate the signature before storing the 3DS-Method status as described above.

Right after submitting the 3DS-Method *(without waiting for the 3DS-Method completion)*, submit a 3DS-Method continue API call to determine what the next step that is required *(as described in the diagram below)*:

- no further action - payment successful / failed
- consumer <--> Issuer interaction needed

In order to submit the 3DS-Method continue call, please make sure to follow the steps, as described below in the diagram:

Submit an API call using HTTP METHOD **PUT** to the URL returned in `threeds_method_continue_url` during the initial transaction request:

- with request headers
  - `Content-Type` - **application/x-www-form-urlencoded**
- with request params
  - `signature` - the same signature used for submitting the 3DS-Method

Please, make sure to submit the 3DS-Method continue with an HTTP PUT request from your backend site, not with an AJAX request from your client side that perfoms a cross-site request. You can still have a custom AJAX request to your own endpoint of the backend API, but the real request to the `threeds_method_continue_url` has to stay hidden. In order to avoid Cross-origin resource sharing issues during the 3DS-Method-Continue submission, make sure to implement an proxy endpoint into your backend site to ensure there will be no Cross-Origin requests. For security reasons, CORS is not allowed and the response header `Access-Control-Allow-Origin` will NOT be sent.

The response of the API call will be transaction response XML indicating what is expected:

- no action, the `status` will be in final state *(authentication completed without friction and authorization has been performed)*
- consumer redirection, the `status` will be **pending_async** *(consumer interaction with the the issuer needed to complete the authentication)*

Please, be aware that this request can take up to 15 seconds to completed. In case of another subsequent request is sent before the 1st one has finished, the API will return HTTP status code **409 Conflict**.

In case of improper **signature** submitted, the API will return HTTP status **400 Bad Request**

For both errors the XML will contain the current status of the transaction (including **in progress**) and state that a reconcile request will have to be sent in order to know what the next step is (approved transaction, declined transaction, challenge requested etc). More detailed information can be found in the diagram below.

Asynchronous 3 D Sv2 Frictionless With 3 Ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4066330000000004</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <threeds_v2_params>
     <threeds_method>
       <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
     </threeds_method>
     <control>
       <device_type>browser</device_type>
       <challenge_window_size>full_screen</challenge_window_size>
       <challenge_indicator>preference</challenge_indicator>
     </control>
     <purchase>
       <category>service</category>
     </purchase>
     <merchant_risk>
       <shipping_indicator>verified_address</shipping_indicator>
       <delivery_timeframe>electronic</delivery_timeframe>
       <reorder_items_indicator>reordered</reorder_items_indicator>
       <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
       <pre_order_date>27-03-2025</pre_order_date>
       <gift_card>true</gift_card>
       <gift_card_count>2</gift_card_count>
     </merchant_risk>
     <card_holder_account>
       <creation_date>27-02-2024</creation_date>
       <update_indicator>more_than_60days</update_indicator>
       <last_change_date>27-11-2024</last_change_date>
       <password_change_indicator>no_change</password_change_indicator>
       <password_change_date>12-02-2025</password_change_date>
       <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
       <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
       <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
       <transactions_activity_previous_year>10</transactions_activity_previous_year>
       <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
       <purchases_count_last_6_months>5</purchases_count_last_6_months>
       <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
       <registration_indicator>30_to_60_days</registration_indicator>
       <registration_date>27-02-2023</registration_date>
     </card_holder_account>
     <browser>
       <accept_header>*/*</accept_header>
       <java_enabled>false</java_enabled>
       <language>en-GB</language>
       <color_depth>24</color_depth>
       <screen_height>900</screen_height>
       <screen_width>1440</screen_width>
       <time_zone_offset>-120</time_zone_offset>
       <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
```

```
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```

Frictionless With 3 Ds Method Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize3d</transaction_type>
  <status>pending_async</status>
  <mode>test</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <threeds_method_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method</threeds_method_url>
  <threeds_method_continue_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method/44177a21403427eb96664a6d7e5d5d48</threeds_method_continue_url>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

```
<!-- Sample HTML for the 3DS-Method submission in an iframe -->
<html>
    <head>
    </head>

    <body onload="submitThreedsMethod()">
        <iframe width="..." height="..." id="threeDSMethodIframe" name="threeDSMethodIframe">
            <html>
              <body>
              </body>
            </html>
        </iframe>

        <form id ="threeDSMethodForm" name="threeDSMethodForm"
              enctype="application/x-www-form-urlencoded;charset=UTF-8"
              style="display: none"
              method="post"
              action="https://staging.gate.e-comprocessing.net/threeds/threeds_method"
              target="threeDSMethodIframe">
            <input type="hidden" name="unique_id" value="44177a21403427eb96664a6d7e5d5d48" />
            <!-- The signature is built as per the above notes for merchant with API password 53ef627e5f9ac4766a8b872a2e0a8548370f0cd2 -->
            <input type="hidden" name="signature" value="85f1b50db01b19b39344c020745ca847813ae3de4ed7dee4db7abdfd1e6650e4565e9331cf5dd84d8bfad8a938c973ddb495b1b9a7d84478d0e22497547550c7" />
        </form>
    </body>

    <script>

        function submitThreedsMethod() {
            threeDSMethodForm = document.getElementById('threeDSMethodForm');
            threeDSMethodForm.submit();
        }

    </script>

</html>
```

```
<!-- 3DS-Method Callback inside the iframe indicating the 3DS-Method completed -->
<!-- Content-Type: application/x-www-form-urlencoded -->
<!-- The signature is built as per the above notes for merchant with API password 53ef627e5f9ac4766a8b872a2e0a8548370f0cd2 -->

POST https://www.example.com/threeds/threeds_method/callback

unique_id=44177a21403427eb96664a6d7e5d5d48
&threeds_method_status=completed
&signature=2e24755dc8622ecf14023852a669fb69a3d61ec2056e60688b8bddf29347f77362b1229c2a98d4a9339e1f685bcb90bd9b0638c025161744f30040db07452c3a
```

```
<!-- 3DS-Method continue API call -->
<!-- Content-Type: application/x-www-form-urlencoded -->
<!-- The signature is built as per the above notes for merchant with API password 53ef627e5f9ac4766a8b872a2e0a8548370f0cd2 -->

curl https://staging.gate.e-comprocessing.net/threeds/threeds_method/44177a21403427eb96664a6d7e5d5d48 \
   -X PUT \
   -H "Content-Type: application/x-www-form-urlencoded" \
   -d signature=85f1b50db01b19b39344c020745ca847813ae3de4ed7dee4db7abdfd1e6650e4565e9331cf5dd84d8bfad8a938c973ddb495b1b9a7d84478d0e22497547550c7
```

3 Ds Method Continue Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize3d</transaction_type>
  <status>approved</status>
  <mode>test</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>true</sent_to_acquirer>
  <scheme_transaction_identifier>019091214161031</scheme_transaction_identifier>
  <scheme_settlement_date>0227</scheme_settlement_date>
  <threeds>
    <eci>05</eci>
  </threeds>
</payment_response>
```

3 Ds Method Continue Declined Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<payment_response>
  <transaction_type>authorize3d</transaction_type>
  <status>declined</status>
  <crypto>true</crypto>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <consumer_id>123456</consumer_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <code>910</code>
  <technical_message>Do not honour</technical_message>
  <message>Transaction declined, please contact support!</message>
  <mode>test</mode>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <threeds>
    <authentication>
      <status_reason_code>01</status_reason_code>
    </authentication>
    <eci>07</eci>
  </threeds>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```


A completion response will be returned by the API providing the status of the transaction. It might also be `approved`, `declined` or `error` depending on the authentication status and the authorization response.

**DIAGRAM**

Merchant
<status>approved</status>
...
</payment_response>
Merchant
Enqueue
an asynchronous notification
about the payment
to notification_url
(HTTP POST)
3DS-Method
required
Merchant
<payment_response>
...
<status>pending_async</status>
<unique_id>...</unique_id>
<threeds_method_url> 3ds_method_url</...>
<threeds_method_continue_url>api_call_continue_url</...>
...
</payment_response>
Gateway
1
1st action:
3DS-Method
submission
2
2nd action
Continue
after
3DS-Method
submission
is initiated
Build an iframe and submit the 3DS-Method inside a form
by targeting the iframe and using HTTP POST
(invisible to the consumer)
<iframe width="xxx" height="xxx" style="..."
id="threeDSMethodIframe" name="threeDSMethodIframe">
<html>
<body></body>
</html>
</iframe>
...
<form name="threeDSMethodForm" id ="threeDSMethodForm" method="post"
enctype="application/x-www-form-urlencoded;charset=UTF-8"
style="display: none"
action="{threeds_method_url}"
target="threeDSMethodIframe">
<input type="hidden" name="unique_id" value="{unique_id}" />
<input type="hidden" name="signature" value="SHA512(...)" />
</form>
...
<script>
function submitThreedsMethod() {
document.getElementById('threeDSMethodForm').submit();
}
</script>
Initiate
3DS-Method
(submitThreedsMethod)
Gateway
Merchant
400
Bad Request
NO
Signature
valid?
YES
<payment_response>
<unique_id>...</unique_id>
<transaction_id>...</transaction_id>
<transaction_type>...</transaction_type>
<status>in_progress / pending_async</status>
<message>Please check signature!</message>
<technical_message>
InputDataInvalidError: 'signature' is incorrect.
</technical_message>
</payment_response>
Merchant
422
Unprocessable Entity
( no response body )
NO
3DS-Method
required?
YES

Gateway
Acquire
3DS-Method Content
(HTTP POST)
Access Control Server (ACS)
3DS-Method content
(HTML + JS)
provided
Merchant (iframe)
Render
3DS-Method content
(HTML + JS)
Gateway
Automatic
3DS-Method form submission
(HTTP POST)
Access Control Server (ACS)
Callback
(HTTP POST)
3DS Server
Callback
(HTTP POST)
Gateway
Transition
3DS-Method status
to Completed
Gateway
Merchant (iframe)
200 OK
( no response body )
NO
callback_url submitted?
YES
No further action needed
( 3DS-Method completed )
Merchant (iframe)
Async Callback
(HTTP POST)
POST {threeds_method_callback_url}?
unique_id={unique-id-here}&threeds_method_status=completed
&signature={SHA512(...)}
Validate
Signature
Merchant (iframe)
Handle 3DS-Method
status
AUTOMATED 3DS-METHOD SUBMISSION
(IFRAME)
Merchant
API call (HTTP PUT)
PUT {threeds_method_continue_url}?signature=
Gateway

{SHA512(...)}
409
Conflict
<payment_response>
<unique_id>...</unique_id>
<transaction_id>...</transaction_id>
<transaction_type>...</transaction_type>
<status>in_progress | pending_async</status>
<technical_message>
Please, reconcile the transaction in order to check its current status.
</technical_message>
</payment_response>
Merchant
YES
Duplicate request?
NO
Send a reconciliation API POST request with unique ID (HTTP POST)
Gateway
<payment_response>
...
<status>pending_async | declined | error</status>
...
</payment_response>
Merchant
Merchant
400
Bad Request
NO
Signature valid?
YES
<payment_response>
<unique_id>...</unique_id>
<transaction_id>...</transaction_id>
<transaction_type>...</transaction_type>
<status>in_progress | pending_async</status>
<message>Please check signature!</message>
<technical_message>
InputDataInvalidError: 'signature' is incorrect.
</technical_message>
</payment_response>
Acquire MPI params
3DS Server
Provide MPI Params
Gateway
Store MPI params
Merchant
<payment_response>
...
<status>error | declined</status>
...
</payment_response>
NO
Authenticated?
YES
Merchant
Enqueue an asynchronous notification about the payment to notification_url (HTTP POST)
Process Transaction
Gateway
Merchant
<payment_response>
...
<status>declined</status>
...
</payment_response>
NO
Transaction successful?
YES
Merchant
<payment_response>
...
<status>approved</status>
...
</payment_response>
Merchant
Enqueue an asynchronous notification about the payment to notification_url (HTTP POST)

**RECONCILE**

Once the transaction reaches the final state, a single reconcile can also be performed to retrieve more detailed information about the 3D transaction. It should include information about the 3DS transaction as described in the reconcile request/response below, on the right.

Reconcile 3 D Transaction By Unique Id Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
 </reconcile>'
```

Successful Reconciliation Of Frictionless 3 D Sv2 Transaction With 3 Ds Method Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>authorize3d</transaction_type>
  <status>approved</status>
  <authorization_code>005645</authorization_code>
  <retrieval_reference_number>016813015184</retrieval_reference_number>
  <payment_account_reference>Q1HJZ8RKA1EBL470G9XYG90R5D3E</payment_account_reference>
  <response_code>00</response_code>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <mode>test</mode>
  <timestamp>2025-02-26T16:01:50Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <card_brand>visa</card_brand>
  <card_number>406633...0004</card_number>
  <card_type>CREDIT</card_type>
  <card_subtype>CARD SUBTYPE</card_subtype>
  <card_issuing_bank>Issuing Bank</card_issuing_bank>
  <card_issuing_country>Exact Issuing country</card_issuing_country>
  <bank_account_number>Bank Account Number</bank_account_number>
  <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
  <sent_to_acquirer>true</sent_to_acquirer>
  <arn>74537605259536043849425</arn>
  <scheme_response_code>00</scheme_response_code>
  <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
  <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
  <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
  <threeds>
    <authentication_flow>frictionless</authentication_flow>
    <threeds_method>
      <status>completed</status>
    </threeds_method>
    <protocol>
      <target_version>2</target_version>
      <concrete_version>2</concrete_version>
      <sub_version>2</sub_version>
    </protocol>
    <eci>05</eci>
  </threeds>
</payment_response>
```

**Successful Reconcile Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to ’true’ if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| **threeds** | | |
| authentication_flow | string(255) | Identifies the concrete authentication flow of the 3DS transaction that it has gone through. It will be included only if the transaction reaches the final state. The possible values for 3DSv2 are **frictionless**, **challenge**. |
| **threeds_method** | | |
| status | string(255) | Identifies the current status of 3DSv2-Method. The possible values are **required**, **in_progress**, **completed**. |
| **protocol** | | |
| target_version | integer | Identifies the requested version of the 3DS authentication protocol to be used. The possible values are **2**. |

| | | |
|---|---|---|
| concrete_version | integer | Identifies the concrete version of the 3DS authentication protocol that the transaction has been processed through. The possible values are **2**. |
| eci | string(2) | See Electronic Commerce Indicator for details |

**NOTIFICATION**

Once the transaction reaches final state, a notification will be sent to the `notification_url` submitted in the initial transaction request. For more information, go to Asynchronous Transactions and Notifications.

Approved Notification Example for frictionless flow with 3DS-Method

```
?transaction_id=119643250547501c79d8295
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=authorize3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=approved
&amount=100
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&eci=05
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&threeds_authentication_flow=frictionless
&threeds_method_status=completed
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

Declined Notification Example for frictionless flow with 3DS-Method

```
?transaction_id=119643250547501c79d8295
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=authorize3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=declined
&amount=100
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&eci=05
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&threeds_authentication_flow=frictionless
&threeds_method_status=completed
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

**3DS Attributes**

| Name | Type | Description |
|---|---|---|
| threeds_authentication_flow | string | Identifies the concrete 3DS authentication flow that the transaction has gone through. It will be available in the notification only if the consumer has finished the 3DS authentication with the issuer. The available values for 3DSv2 are frictionless and challenge. |
| threeds_method_status | string | Identifies the status of the 3DS-Method in the scope of 3DSv2 authentication protocol. The possible values are **required**, **in_progress** and **completed**. |
| threeds_target_protocol_version | string(1) | Identifies the 3DS protocol that has been enforced. The possible values are **2**. |
| threeds_concrete_protocol_version | string(1) | Identifies the concrete 3DS protocol version that the transaction has gone through. The possible values are **2**. |
| threeds_protocol_sub_version | string(1) | Identifies the 3DS protocol sub-version that the transaction has gone through. The possible values are **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**. |
| threeds_authentication_status_reason_code | string(2) | See Status Reason Code for details. |

Please have in mind, the above 3DS related params will be available for all transactions supporting 3DS in async workflow: Authorize3d, Sale3d, or InitRecurringSale3d. For more information about the 3DS transactions, go to 3DS Card.

**CHALLENGE**

If the ACS determines that further Cardholder interaction is required to complete the authentication, the Frictionless Flow transitions into the Challenge Flow. For example, a challenge may be necessary because the transaction is deemed high-risk, is above certain thresholds, or requires a higher level of authentication due to country mandates (or regulations).

The Challenge Flow occurs when the issuer assesses the risk of the transaction during the frictionless flow and determines that the transaction requires additional cardholder authentication. The frictionless flow transitions into the challenge flow

In order to enforce using the 3DSv2 authentication protocol, make sure to **include** the **threeds_v2_params** in the transaction request.

An exemption from Strong Customer Authentication (SCA) can be requested by submitting an **exemption** with `low_risk` under SCA params.

In case the issuer accepts the exemption, a step up in the authentication flow might not be required because the transaction's risk analysis has already been performed by acquirer.

Note, the requested exemption might not be accepted due to internal risk validations.

For example, to be able to utilize the low risk exemption, the BIN country of the card must be part of the European Economic Area (EEA).

Furthermore, the acquirer could accept the merchant low-risk exemption request only if the transaction amount does not exceed the acquirer low-risk exemption threshold.

Finally, the ACS might not acknowledge the merchant/acquirer's exemption request and may still require a step up in the cardholder authentication.

Asynchronous 3 D Sv2 Challenge Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
```

```
  <transaction_type>init_recurring_sale3d</transaction_type>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4918190000000002</card_number>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <cvv>834</cvv>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <recurring>
      <expiration_date>27-08-2025</expiration_date>
      <frequency>30</frequency>
    </recurring>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```


Challenge Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>init_recurring_sale3d</transaction_type>
  <status>pending_async</status>
  <mode>test</mode>
  <transaction_id>1196432505475 01c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/threeds/authentication/44177a21403427eb96664a6d7e5d5d48</redirect_url>
  <redirect_url_type>3ds_v2_challenge</redirect_url_type>
  <timestamp>2025-02-26T16:01:51Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```


**DIAGRAM**

sale3d, init_recurring_sale3d)
POST/#API_URL/process/TERMINAL_TOKEN
<payment_transaction>
<transaction_type>authorize3d | sale3d |...</...>
...
<return_success_url>...</return_success_url>
<return_failure_url>...</return_failure_url>
<notification_url>...</notification_url>
<threeds_v2_params>
...
</threeds_v2_params>
</payment_transaction>
Merchant
Gateway
Card valid?
NO
YES
<payment_response>
...
<status>error</status>
...
</payment_response>
Merchant
Acquire
Enrollment status
Enqueue
an asynchronous notification
about the payment
to notification_url
(HTTP POST)
Merchant
3DS Server
Enrollment status
provided
Enrolled?
NO
YES
Authorization
allowed?
NO
YES
<payment_response>
...
<status>error | declined</status>
...
</payment_response>
Merchant
Process
Transaction
Enqueue
an asynchronous notification
about the payment
to notification_url
(HTTP POST)
Merchant
Gateway
Transaction
successful?
NO
YES
<payment_response>
...
<status>declined</status>
...
</payment_response>
Merchant
<payment_response>
...
<status>approved</status>
...
</payment_response>
Merchant
Enqueue
an asynchronous notification
about the payment
to notification_url
(HTTP POST)
Merchant
3DS-Challenge
required
<payment_response>
...
<status>pending_async</status>
<redirect_url>3ds_challenge_url</redirect_url>
<redirect_url_type>3ds_v2_challenge</...>
...
</payment_response>
Gateway
Handle response
Merchant
Redirect consumer to
redirect_url
(HTTP GET)
Gateway
Acquire
Challenge Form
(HTTP POST)

Access Control Server (ACS)
Challenge Form provided
Gateway
Render Challenge
Consumer
Solve Challenge (strong authentication)
Consumer
Consumer interacts with the issuer to complete the strong 3DS authentication
Access Control Server (ACS)
Solve Challenge (strong authentication)
3DS-Challenge
Verify Challenge
Consumer
Render another Challenge
YES
Challenge again?
NO
Prepare Callback (Challenge finished)
Access Control Server (ACS)
Callback (HTTP POST)
Gateway
Acquire MPI params
3DS Server
MPI params provided
Gateway
Store MPI params
Merchant
Back to merchant
Consumer
Redirect consumer to return_failure_url (HTTP GET)
NO
Authenticated?
YES
Merchant
Enqueue an asynchronous notification about the payment to notification_url (HTTP POST)
Process Transaction
Gateway

**RECONCILE**

Once the transaction reaches the final state, a single reconcile can also be performed to retrieve more detailed information about the 3D transaction. It should include information about the 3DS transaction as described in the reconcile request/response below, on the right.

Reconcile 3 D Transaction By Unique Id Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
 </reconcile>'
```

Successful Reconciliation Of 3 D Sv2 Transaction With Challenge Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>init_recurring_sale3d</transaction_type>
   <status>approved</status>
   <authorization_code>005645</authorization_code>
   <retrieval_reference_number>016813015184</retrieval_reference_number>
   <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
   <response_code>00</response_code>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <mode>test</mode>
   <timestamp>2025-02-26T16:01:51Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
   <card_brand>visa</card_brand>
   <card_number>491819...0002</card_number>
   <card_type>CREDIT</card_type>
   <card_subtype>CARD SUBTYPE</card_subtype>
   <card_issuing_bank>Issuing Bank</card_issuing_bank>
   <card_issuing_country>Exact Issuing country</card_issuing_country>
   <bank_account_number>Bank Account Number</bank_account_number>
   <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
   <sent_to_acquirer>true</sent_to_acquirer>
   <arn>74537605259536043849425</arn>
   <scheme_response_code>00</scheme_response_code>
   <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
   <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
   <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
   <threeds>
     <authentication_flow>challenge</authentication_flow>
     <protocol>
       <target_version>2</target_version>
       <concrete_version>2</concrete_version>
       <sub_version>2</sub_version>
     </protocol>
     <eci>05</eci>
   </threeds>
 </payment_response>
```

**Successful Reconcile Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |

| | | |
|---|---|---|
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to 'true' if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| **threeds** | | |
| authentication_flow | string(255) | Identifies the concrete authentication flow of the 3DS transaction that it has gone through. It will be included only if the transaction reaches the final state. The possible values for 3DSv2 are **frictionless**, **challenge**. |
| **protocol** | | |
| target_version | integer | Identifies the requested version of the 3DS authentication protocol to be used. The possible values are **2**. |
| concrete_version | integer | Identifies the concrete version of the 3DS authentication protocol that the transaction has been processed through. The possible values are **2**. |
| eci | string(2) | See Electronic Commerce Indicator for details |

#### NOTIFICATION

Once the transaction reaches final state, a notification will be sent to the `notification_url` submitted in the initial transaction request. For more information, go to Asynchronous Transactions and Notifications.

Approved Notification Example for challenge flow

```
?transaction_id=119643250547501c79d8295
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=init_recurring_sale3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=approved
&amount=100
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&eci=05
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&threeds_authentication_flow=challenge
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

Declined Notification Example for challenge flow

```
?transaction_id=119643250547501c79d8295
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=init_recurring_sale3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=declined
&amount=100
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&eci=05
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&threeds_authentication_flow=challenge
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

**3DS Attributes**

| Name | Type | Description |
|---|---|---|
| threeds_authentication_flow | string | Identifies the concrete 3DS authentication flow that the transaction has gone through. It will be available in the notification only if the consumer has finished the 3DS authentication with the issuer. The available values for 3DSv2 are frictionless and challenge. |
| threeds_method_status | string | Identifies the status of the 3DS-Method in the scope of 3DSv2 authentication protocol. The possible values are **required**, **in_progress** and **completed**. |
| threeds_target_protocol_version | string(1) | Identifies the 3DS protocol that has been enforced. The possible values are **2**. |
| threeds_concrete_protocol_version | string(1) | Identifies the concrete 3DS protocol version that the transaction has gone through. The possible values are **2**. |
| threeds_protocol_sub_version | string(1) | Identifies the 3DS protocol sub-version that the transaction has gone through. The possible values are **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**. |
| threeds_authentication_status_reason_code | string(2) | See Status Reason Code for details. |

Please have in mind, the above 3DS related params will be available for all transactions supporting 3DS in async workflow: Authorize3d, Sale3d, or InitRecurringSale3d. For more information about the 3DS transactions, go to 3DS Card.

#### CHALLENGE WITH 3DSECURE METHOD

If the ACS determines that further Cardholder interaction is required to complete the authentication, the Frictionless Flow transitions into the Challenge Flow. For example, a challenge may be necessary because the transaction is deemed high-risk, is above certain thresholds, or requires a higher level of authentication due to country mandates (or regulations).

The Challenge Flow occurs when the issuer assesses the risk of the transaction during the frictionless flow and determines that the transaction requires additional cardholder authentication. The frictionless flow transitions into the challenge flow

In order to enforce using the 3DSv2 authentication protocol, make sure to **include** the **threeds_v2_params** in the transaction request.

An exemption from Strong Customer Authentication (SCA) can be requested by submitting an **exemption** with `low_risk` under SCA params.

In case the issuer accepts the exemption, a step up in the authentication flow might not be required because the transaction's risk analysis has already been performed by acquirer.

Note, the requested exemption might not be accepted due to internal risk validations.

For example, to be able to utilize the low risk exemption, the BIN country of the card must be part of the European Economic Area (EEA).

Furthermore, the acquirer could accept the merchant low-risk exemption request only if the transaction amount does not exceed the acquirer low-risk exemption threshold.

Finally, the ACS might not acknowledge the merchant/acquirer's exemption request and may still require a step up in the cardholder authentication.

**Asynchronous 3DS Method submission**

A link between the customer's browser and the card issuer must be opened with a hidden iframe. It is used for the card issuer to load JavaScript which gathers device information to be returned to the card issuer. The next step after initiating the iframe, is to submit an API call to the `threeds_method_continue_url` using **HTTP PUT**, to retrieve the next step in the authentication. The API request won't require HTTP basic authentication, but a proper **signature** needs to be included to prove the authenticity of the request. The response of this API call will be the same as the normal 3D transaction processing API response, but with the only difference that an additional interaction with the issuer might be requested.

The asynchronous submission of the 3DS-Method might look difficult to achieve, but mitigates the risk of potential transaction processing interruptions, because the consumer redirection does not depend on a successful 3DS-Method submission. A continuation of the 3DS-Method might be requested regardless of the 3DS-Method submission result. It's a responsibility of the ACS then to take the appropriate decision how to continue with the authentication.

In order to simulate this authentication flow, use a test card 4938730000000001 for challenge flow that requires 3DS-Method and submit a 3DSv2 transaction in asynchronous workflow by including the `threeds_v2_params`.

The response of the API will indicate that further action is required:

- `status` - **pending_async**
- `threeds_method_url` - the URL action where the 3DS-Method needs to be submitted using HTTP POST
- `threeds_method_continue_url` - API endpoint that accepts HTTP PUT requests with a signature and returns transaction API response identifying what the next step is *(transaction completed or consumer interaction is needed)*

In order to submit a 3DS-Method, you need to create a hidden iframe in the consumer browser (client side) with an `html` and `body` tags as described below, on the right side and create a hidden **HTML form** that:

- targets the iframe
- uses HTTP METHOD **POST** - `method="post"`
- has an **action** equivalent to the value of `threeds_method_url`, received from the response of the initial transaction request
- has 2 hidden inputs
  - `unique_id` - equivalent to the value of the `unique_id`, received from the response of the initial transaction request
  - `signature` - **SHA512** of a concatenated string (**unique_id**, **amount**, **timestamp**, **merchant_api_password**), where **unique_id**, **amount**, **timestamp** can be taken from the response of the initial transaction request and **merchant_api_password** is the password used for HTTP Basic authentication to the API during the initial transaction request
- submit the HTML form in the background using JavaScript

Once the 3DS-Method submission is initiated, a callback via HTTP POST will be done inside the iframe when the 3DS-Method reaches the final state. The 3DS-Method callback will be sent to the `callback_url` (submitted in the 3DSv2 request params as described in the diagram below) and will be constructed as described below:

- with request headers
  - `Content-Type` - **application/x-www-form-urlencoded**
- with POST request params
  - `unique_id` - the exact **unique_id** of the transaction in the initial transaction request
  - `threeds_method_status` - the status of the 3DS-Method submission, expect a value of **completed**
  - `signature` - **SHA512** of a concatenated string with the values of **unique_id**, **threeds_method_status**, and **merchant_api_password** where **unique_id** and **threeds_method_status** are POST params and **merchant_api_password** is the password used for HTTP HTTP Basic authentication to the API during the initial transaction request

The callback above can be handled *(optional)* in order to get informed of the 3DS-Method status, whether it has completed or not.

The 3DS-Method callback is sent asynchronously as shown in the diagram.

In case you have implemented the 3DS-Method callback handler (it's not required), make sure you validate the signature before storing the 3DS-Method status as described above.

Right after submitting the 3DS-Method *(without waiting for the 3DS-Method completion)*, submit a 3DS-Method continue API call to determine what the next step that is required *(as described in the diagram below)*:

- no further action - payment successful / failed
- consumer <--> Issuer interaction needed

In order to submit the 3DS-Method continue call, please make sure to follow the steps, as described below in the diagram:

Submit an API call using HTTP METHOD **PUT** to the URL returned in `threeds_method_continue_url` during the initial transaction request:

- with request headers
  - `Content-Type` - **application/x-www-form-urlencoded**
- with request params
  - `signature` - the same signature used for submitting the 3DS-Method

Please, make sure to submit the 3DS-Method continue with an HTTP PUT request from your backend site, not with an AJAX request from your client side that perfoms a cross-site request. You can still have a custom AJAX request to your own endpoint of the backend API, but the real request to the `threeds_method_continue_url` has to stay hidden. In order to avoid Cross-origin resource sharing issues during the 3DS-Method-Continue submission, make sure to implement an proxy endpoint into your backend site to ensure there will be no Cross-Origin requests. For security reasons, CORS is not allowed and the response header `Access-Control-Allow-Origin` will NOT be sent.

The response of the API call will be transaction response XML indicating what is expected:

- no action, the `status` will be in final state *(authentication completed without friction and authorization has been performed)*
- consumer redirection, the `status` will be **pending_async** *(consumer interaction with the the issuer needed to complete the authentication)*

Please, be aware that this request can take up to 15 seconds to completed. In case of another subsequent request is sent before the 1st one has finished, the API will return HTTP status code **409 Conflict**.

In case of improper **signature** submitted, the API will return HTTP status **400 Bad Request**

For both errors the XML will contain the current status of the transaction (including **in progress**) and state that a reconcile request will have to be sent in order to know what the next step is (approved transaction, declined transaction, challenge requested etc). More detailed information can be found in the diagram below.

Asynchronous 3 D Sv2 Challenge With 3 Ds Method Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>init_recurring_sale3d</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4938730000000001</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
```

```
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <threeds_v2_params>
    <threeds_method>
      <callback_url>https://www.example.com/threeds/threeds_method/callback</callback_url>
    </threeds_method>
    <control>
      <device_type>browser</device_type>
      <challenge_window_size>full_screen</challenge_window_size>
      <challenge_indicator>preference</challenge_indicator>
    </control>
    <purchase>
      <category>service</category>
    </purchase>
    <recurring>
      <expiration_date>27-08-2025</expiration_date>
      <frequency>30</frequency>
    </recurring>
    <merchant_risk>
      <shipping_indicator>verified_address</shipping_indicator>
      <delivery_timeframe>electronic</delivery_timeframe>
      <reorder_items_indicator>reordered</reorder_items_indicator>
      <pre_order_purchase_indicator>merchandise_available</pre_order_purchase_indicator>
      <pre_order_date>27-03-2025</pre_order_date>
      <gift_card>true</gift_card>
      <gift_card_count>2</gift_card_count>
    </merchant_risk>
    <card_holder_account>
      <creation_date>27-02-2024</creation_date>
      <update_indicator>more_than_60days</update_indicator>
      <last_change_date>27-11-2024</last_change_date>
      <password_change_indicator>no_change</password_change_indicator>
      <password_change_date>12-02-2025</password_change_date>
      <shipping_address_usage_indicator>current_transaction</shipping_address_usage_indicator>
      <shipping_address_date_first_used>22-02-2025</shipping_address_date_first_used>
      <transactions_activity_last_24_hours>2</transactions_activity_last_24_hours>
      <transactions_activity_previous_year>10</transactions_activity_previous_year>
      <provision_attempts_last_24_hours>1</provision_attempts_last_24_hours>
      <purchases_count_last_6_months>5</purchases_count_last_6_months>
      <suspicious_activity_indicator>no_suspicious_observed</suspicious_activity_indicator>
      <registration_indicator>30_to_60_days</registration_indicator>
      <registration_date>27-02-2023</registration_date>
    </card_holder_account>
    <browser>
      <accept_header>*/*</accept_header>
      <java_enabled>false</java_enabled>
      <language>en-GB</language>
      <color_depth>24</color_depth>
      <screen_height>900</screen_height>
      <screen_width>1440</screen_width>
      <time_zone_offset>-120</time_zone_offset>
      <user_agent>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36</user_agent>
    </browser>
    <sdk>
      <interface>native</interface>
      <ui_types>
        <ui_type>multi_select</ui_type>
      </ui_types>
      <application_id>fc1650c0-5778-0138-8205-2cbc32a32d65</application_id>
      <encrypted_data>encrypted-data-here</encrypted_data>
      <ephemeral_public_key_pair>public-key-pair</ephemeral_public_key_pair>
      <max_timeout>10</max_timeout>
      <reference_number>sdk-reference-number-here</reference_number>
    </sdk>
  </threeds_v2_params>
  <sca_params>
    <exemption>low_risk</exemption>
  </sca_params>
</payment_transaction>'
```


Challenge With 3 Ds Method Response

Response


```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>init_recurring_sale3d</transaction_type>
  <status>pending_async</status>
  <mode>test</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <threeds_method_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method</threeds_method_url>
  <threeds_method_continue_url>https://staging.gate.e-comprocessing.net/threeds/threeds_method/44177a21403427eb96664a6d7e5d5d48</threeds_method_continue_url>
  <timestamp>2025-02-26T16:01:51Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```


```
<!-- Sample HTML for the 3DS-Method submission in an iframe -->
<html>
    <head>
    </head>

    <body onload="submitThreedsMethod()">
        <iframe width="..." height="..." id="threeDSMethodIframe" name="threeDSMethodIframe">
            <html>
              <body>
              </body>
            </html>
        </iframe>

        <form id ="threeDSMethodForm" name="threeDSMethodForm"
              enctype="application/x-www-form-urlencoded;charset=UTF-8"
              style="display: none"
              method="post"
              action="https://staging.gate.e-comprocessing.net/threeds/threeds_method"
              target="threeDSMethodIframe">
            <input type="hidden" name="unique_id" value="44177a21403427eb96664a6d7e5d5d48" />
            <!-- The signature is built as per the above notes for merchant with API password 53ef627e5f9ac4766a8b872a2e0a8548370f0cd2 -->
            <input type="hidden" name="signature" value="9e8e67a78fcd2f05647e1cef8d29c794ff559c2ff863e3ce92290d44ee81f82977045a6f8de74b3dbb811c4c0c39a1d169690bc79a494dd8f546717474d73b85" />
        </form>
    </body>

    <script>

        function submitThreedsMethod() {
            threeDSMethodForm = document.getElementById('threeDSMethodForm');
```

```
            threeDSMethodForm.submit();
        }

    </script>

</html>
```

```
<!-- 3DS-Method Callback inside the iframe indicating the 3DS-Method completed -->
<!-- Content-Type: application/x-www-form-urlencoded -->
<!-- The signature is built as per the above notes for merchant with API password 53ef627e5f9ac4766a8b872a2e0a8548370f0cd2 -->

POST https://www.example.com/threeds/threeds_method/callback

unique_id=44177a21403427eb96664a6d7e5d5d48
&threeds_method_status=completed
&signature=2e24755dc8622ecf14023852a669fb69a3d61ec2056e60688b8bddf29347f77362b1229c2a98d4a9339e1f685bcb90bd9b0638c025161744f30040db07452c3a
```

```
<!-- 3DS-Method continue API call -->
<!-- Content-Type: application/x-www-form-urlencoded -->
<!-- The signature is built as per the above notes for merchant with API password 53ef627e5f9ac4766a8b872a2e0a8548370f0cd2 -->

curl https://staging.gate.e-comprocessing.net/threeds/threeds_method/44177a21403427eb96664a6d7e5d5d48 \
   -X PUT \
   -H "Content-Type: application/x-www-form-urlencoded" \
   -d signature=9e8e67a78fcd2f05647e1cef8d29c794ff559c2ff863e3ce92290d44ee81f82977045a6f8de74b3dbb811c4c0c39a1d169690bc79a494dd8f546717474d73b85
```

3 Ds Method Continue Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payment_response>
  <transaction_type>init_recurring_sale3d</transaction_type>
  <status>pending_async</status>
  <mode>test</mode>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
  <redirect_url>https://staging.gate.e-comprocessing.net/threeds/authentication/44177a21403427eb96664a6d7e5d5d48</redirect_url>
  <redirect_url_type>3ds_v2_challenge</redirect_url_type>
  <timestamp>2025-02-26T16:01:51Z</timestamp>
  <descriptor>Descriptor one</descriptor>
  <amount>100</amount>
  <currency>USD</currency>
  <sent_to_acquirer>false</sent_to_acquirer>
</payment_response>
```

The consumer needs to be redirected to the Challenge `redirect_url` URL to complete the authentication with the ACS. Once the consumer completes all the challenges of the ACS provider, will be redirected either to `return_success_url` or `return_failure_url` depending on the challenge authentication status and authorization response as described in the diagram below.

**DIAGRAM**

Merchant
Enqueue
an asynchronous notification
about the payment
to notification_url
(HTTP POST)
Process
Transaction
Gateway
Merchant
<payment_response>
...
<status>declined</status>
...
</payment_response>
NO
Transaction
successful?
YES
Merchant
<payment_response>
...
<status>approved</status>
...
</payment_response>
Merchant
Enqueue
an asynchronous notification
about the payment
to notification_url
(HTTP POST)
3DS-Method
required
Merchant
<payment_response>
...
<status>pending_async</status>
<unique_id>...</unique_id>
<threeds_method_url> 3ds_method_url</...>
<threeds_method_continue_url>api_call_continue_url</...>
...
</payment_response>
Gateway
1
1st action:
3DS-Method
submission
2
2nd action
Continue
after
3DS-Method
submission
is initiated
Build an iframe and submit the 3DS-Method inside a form
by targeting the iframe and using HTTP POST
(invisible to the consumer)
<iframe width="xxx" height="xxx" style="..."
id="threeDSMethodIframe" name="threeDSMethodIframe">
<html>
<body></body>
</html>
</iframe>
...
<form name="threeDSMethodForm" id ="threeDSMethodForm" method="post"
enctype="application/x-www-form-urlencoded;charset=UTF-8"
style="display: none"
action="{threeds_method_url}"
target="threeDSMethodIframe">
<input type="hidden" name="unique_id" value="{unique_id}" />
<input type="hidden" name="signature" value="SHA512(...)" />
</form>
...
<script>
function submitThreedsMethod() {
document.getElementById('threeDSMethodForm').submit();
}
</script>
Initiate
3DS-Method
(submitThreedsMethod)
Gateway

Merchant
400
Bad Request
NO
Signature valid?
YES
<payment_response>
<unique_id>...</unique_id>
<transaction_id>...</transaction_id>
<transaction_type>...</transaction_type>
<status>in_progress | pending_async</status>
<message>Please check signature!</message>
<technical_message>
InputDataInvalidError: 'signature' is incorrect.
</technical_message>
</payment_response>
Merchant
422
Unprocessable Entity
( no response body )
NO
3DS-Method required?
YES
Gateway
Acquire
3DS-Method Content
(HTTP POST)
Access Control Server (ACS)
3DS-Method content
(HTML + JS)
provided
Merchant (iframe)
Render
3DS-Method content
(HTML + JS)
Gateway
Automatic
3DS-Method form submission
(HTTP POST)
Access Control Server (ACS)
Callback
(HTTP POST)
3DS Server
Callback
(HTTP POST)
Gateway
Transition
3DS-Method status
to Completed
Gateway
Merchant (iframe)
200 OK
( no response body )
NO
callback_url submitted?
YES
No further action needed
( 3DS-Method completed )
Merchant (iframe)
Callback
(HTTP POST)

POST {threeds_method_callback_url}?unique_id={unique-id-here}&threeds_method_status=completed&signature={SHA512(...)}
Validate Signature
Merchant (iframe)
Handle 3DS-Method status
AUTOMATED 3DS-METHOD SUBMISSION (IFRAME)
Merchant
API call (HTTP PUT)
PUT {threeds_method_continue_url}?signature={SHA512(...)}
Gateway
409 Conflict
<payment_response>
<unique_id>...</unique_id>
<transaction_id>...</transaction_id>
<transaction_type>...</transaction_type>
<status>in_progress | pending_async</status>
<technical_message>
Please, reconcile the transaction in order to check its current status.
</technical_message>
</payment_response>
YES
Duplicate request?
NO
Send a reconciliation API POST request with unique ID (HTTP POST)
Gateway
<payment_response>
...
<status>pending_async | declined | error</status>
...
</payment_response>
Merchant
Merchant
400 Bad Request
NO
Signature valid?
YES
<payment_response>
<unique_id>...</unique_id>
<transaction_id>...</transaction_id>
<transaction_type>...</transaction_type>
<status>in_progress | pending_async</status>
<message>Please check signature!</message>
<technical_message>
InputDataInvalidError: 'signature' is incorrect.
</technical_message>
</payment_response>
Acquire MPI params
3DS Server
Challenge required
Gateway
<payment_response>
...
<status>pending_async</status>
<redirect_url>3ds_v2_challenge_url</redirect_url>
<redirect_url_type>3ds_v2_challenge</redirect_url_type>
...
</payment_response>
Merchant
Handle response
Merchant
Redirect consumer to redirect_url (HTTP GET)
Gateway
Acquire Challenge Form (HTTP POST)
Access Control Server (ACS)

Challenge Form provided
Gateway
Render Challenge
Consumer
Solve Challenge (strong authentication)
Consumer
Consumer interacts with the issuer to complete the strong 3DS authentication
Access Control Server (ACS)
Verify Challenge
3DS-Challenge
Challenge again?
YES
Render another Challenge
Consumer
Solve Challenge (strong authentication)
NO
Prepare Callback (Challenge finished)
Access Control Server (ACS)
Callback (HTTP POST)
Gateway
Acquire MPI params
3DS Server
MPI Params provided
Gateway
Store MPI params
Authenticated?
NO
Redirect consumer to return_failure_url (HTTP GET)
Consumer
Back to merchant
Merchant
YES
Process Transaction
Enqueue an asynchronous notification about the payment to notification_url (HTTP POST)
Merchant
Gateway
Transaction
NO
Redirect consumer to return_failure_url
Consumer
Back to
Merchant

**RECONCILE**

Once the transaction reaches the final state, a single reconcile can also be performed to retrieve more detailed information about the 3D transaction. It should include information about the 3DS transaction as described in the reconcile request/response below, on the right.

Reconcile 3 D Transaction By Unique Id Request

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/reconcile/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <reconcile>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
 </reconcile>'
```

Successful Reconciliation Of 3 D Sv2 Transaction With Challenge And 3 Ds Method Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_type>init_recurring_sale3d</transaction_type>
   <status>approved</status>
   <authorization_code>005645</authorization_code>
   <retrieval_reference_number>016813015184</retrieval_reference_number>
   <payment_account_reference>Q1HJZ28RKA1EBL470G9XYG90R5D3E</payment_account_reference>
   <response_code>00</response_code>
   <unique_id>44177a21403427eb96664a6d7e5d5d48</unique_id>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <mode>test</mode>
   <timestamp>2025-02-26T16:01:51Z</timestamp>
   <descriptor>Descriptor one</descriptor>
   <amount>100</amount>
   <currency>USD</currency>
   <card_brand>visa</card_brand>
   <card_number>493873...0001</card_number>
   <card_type>CREDIT</card_type>
   <card_subtype>CARD SUBTYPE</card_subtype>
   <card_issuing_bank>Issuing Bank</card_issuing_bank>
   <card_issuing_country>Exact Issuing country</card_issuing_country>
   <bank_account_number>Bank Account Number</bank_account_number>
   <bank_identifier_code>Bank Identifier Code</bank_identifier_code>
   <sent_to_acquirer>true</sent_to_acquirer>
   <arn>74537605259536043849425</arn>
   <scheme_response_code>00</scheme_response_code>
   <scheme_transaction_link_id>TLINKIDENTIFIER4521412</scheme_transaction_link_id>
   <scheme_max_settlement_date>2024-07-31</scheme_max_settlement_date>
   <scheme_authentication_data_quality>true</scheme_authentication_data_quality>
   <threeds>
     <authentication_flow>challenge</authentication_flow>
     <threeds_method>
       <status>completed</status>
     </threeds_method>
     <protocol>
       <target_version>2</target_version>
       <concrete_version>2</concrete_version>
       <sub_version>2</sub_version>
     </protocol>
     <eci>05</eci>
   </threeds>
 </payment_response>
```

**Successful Reconcile Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| transaction_type | string(255) | The transaction type |
| status | string(255) | Status of the transaction, see states |
| transaction_id | string(255) | Unique transaction id defined by merchant |
| unique_id | string(32) | Unique id defined by gate (must later be used if capturing, voiding or refunding a transaction) |
| moto | 'true' | Signifies whether a MOTO (mail order telephone order) transaction is performed. Contact tech support for more details. |
| avs_response_code | string(255) | Generated by the card network on trying to match the billing address when performing the address verification. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| avs_response_text | string(255) | Gives the human response text for the AVS response code above. Optional, returned if config is enabled and acquirer supports it. Check AVS Status Codes for details. |
| cvv_result_code | string(1) | Card Verification Value response code. Optional, returned only if acquirer supports it |
| authorization_code | string(6) | Generated by the card network when an authorisation has occurred, used to identify that auth. Consists of 6 alphanumeric chars |
| retrieval_reference_number | string(255) | A reference number used for tracking all messages related to a given cardholder transaction returned by some acquirers. |
| response_code | string(2) | Defines the result of a transaction with the acquirer, the status of a message or some action taken or required. Contact tech-support@emerchantpay.com for more details. |
| mode | string(4) | Mode of the transaction’s terminal, can be **test** or **live** |
| timestamp | string(255) | Time when the transaction was processed in ISO 8601 Combined date and time e.g. 2007-08- 30T17:46:11Z |

| | | |
|---|---|---|
| descriptor | string(255) | Static descriptor MID info as configured on the gateway |
| amount | integer | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | string(255) | Currency code in ISO 4217 |
| partial_approval | string(4) | Optional, set to 'true' if partial approval happened. Partially approved amount is then in the amount field. Check Partial Approvals for details |
| sent_to_acquirer | string(255) | "true" or "false" |
| **threeds** | | |
| authentication_flow | string(255) | Identifies the concrete authentication flow of the 3DS transaction that it has gone through. It will be included only if the transaction reaches the final state. The possible values for 3DSv2 are **frictionless**, **challenge**. |
| **threeds_method** | | |
| status | string(255) | Identifies the current status of 3DSv2-Method. The possible values are **required**, **in_progress**, **completed**. |
| **protocol** | | |
| target_version | integer | Identifies the requested version of the 3DS authentication protocol to be used. The possible values are **2**. |
| concrete_version | integer | Identifies the concrete version of the 3DS authentication protocol that the transaction has been processed through. The possible values are **2**. |
| eci | string(2) | See Electronic Commerce Indicator for details |

#### NOTIFICATION

Once the transaction reaches final state, a notification will be sent to the `notification_url` submitted in the initial transaction request. For more information, go to Asynchronous Transactions and Notifications.

Approved Notification Example for challenge flow with 3DS-Method

```
?transaction_id=119643250547501c79d8295
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=init_recurring_sale3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=approved
&amount=100
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&eci=05
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&threeds_authentication_flow=challenge
&threeds_method_status=completed
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

Declined Notification Example for challenge flow with 3DS-Method

```
?transaction_id=119643250547501c79d8295
&unique_id=44177a21403427eb96664a6d7e5d5d48
&transaction_type=init_recurring_sale3d
&terminal_token=394f2ebc3646d3c017fa1e1cbc4a1e20
&status=declined
&amount=100
&signature=088e16a1019277b15d58faf0541e11910eb756f6
&eci=05
&avs_response_code=5I
&avs_response_text=Response+provided+by+issuer+processor%3B+Address+information+not+verified
&cvv_result_code=M
&authorization_code=005645
&retrieval_reference_number=016813015184
&payment_account_reference=50019P9LBXOLHN9G7QMU5VN520YSY
&scheme_transaction_link_id=TLINKIDENTIFIER4521412
&scheme_pan_indicator=V
&scheme_pan_tail=0013
&scheme_max_settlement_date=2024-07-31
&scheme_authentication_data_quality=true
&threeds_authentication_flow=challenge
&threeds_method_status=completed
&threeds_target_protocol_version=2
&threeds_concrete_protocol_version=2
&threeds_protocol_sub_version=2
```

**3DS Attributes**

| Name | Type | Description |
|---|---|---|
| threeds_authentication_flow | string | Identifies the concrete 3DS authentication flow that the transaction has gone through. It will be available in the notification only if the consumer has finished the 3DS authentication with the issuer. The available values for 3DSv2 are frictionless and challenge. |
| threeds_method_status | string | Identifies the status of the 3DS-Method in the scope of 3DSv2 authentication protocol. The possible values are **required**, **in_progress** and **completed**. |
| threeds_target_protocol_version | string(1) | Identifies the 3DS protocol that has been enforced. The possible values are **2**. |
| threeds_concrete_protocol_version | string(1) | Identifies the concrete 3DS protocol version that the transaction has gone through. The possible values are **2**. |
| threeds_protocol_sub_version | string(1) | Identifies the 3DS protocol sub-version that the transaction has gone through. The possible values are **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**. |
| threeds_authentication_status_reason_code | string(2) | See Status Reason Code for details. |

Please have in mind, the above 3DS related params will be available for all transactions supporting 3DS in async workflow: Authorize3d, Sale3d, or InitRecurringSale3d. For more information about the 3DS transactions, go to 3DS Card.

#### STATUS REASON CODE

The authentication status reason code is a predefined code as per the EMVCo specification and provides further information about the reason for the failed/declined 3DS authentication or the error that has occurred while trying to authenticate the cardholder.

| Code | Description |
|---|---|
| 01 | Card authentication failed |
| 02 | Unknown Device |
| 03 | Unsupported Device |

| | |
|---|---|
| 04 | Exceeds authentication frequency limit |
| 05 | Expired card |
| 06 | Invalid card number |
| 07 | Invalid transaction |
| 08 | No Card record |
| 09 | Security failure |
| 10 | Stolen card |
| 11 | Suspected fraud |
| 12 | Transaction not permitted to cardholder |
| 13 | Cardholder not enrolled in service |
| 14 | Transaction timed out at the ACS |
| 15 | Low confidence |
| 16 | Medium confidence For 01-PA, required if the Transaction Status field = N, U, or R. For 02-NPA, Conditional asdefined by the DS. |
| 17 | High confidence |
| 18 | Very High confidence |
| 19 | Exceeds ACS maximum challenges |
| 20 | Non-Payment transaction not supported |
| 21 | 3RI transaction not supported |
| 22 | ACS technical issue |
| 23 | Decoupled Authentication required by ACS but not requested by 3DS Requestor |
| 24 | 3DS Requestor Decoupled Max Expiry Time exceeded |
| 25 | Decoupled Authentication was provided insufficient time to authenticate cardholder. ACS will not make attempt |
| 26 | Authentication attempted but not performed by the cardholder |
| 27–79 | Reserved for EMVCo future use (values invalid until defined by EMVCo) |
| 87 | Transaction is excluded from Attempts Processing (includes non-reloadable pre-paid cards and Non-Payments (NPA)) (Visa only) |

**TESTING**

| Scenario | 3DSecure Method | 3DSecure Challenge | Result | Card Number | Note |
|---|---|---|---|---|---|
| Frictionless | - | - | Authenticated | 4012000000060085 | |
| Frictionless | - | - | Authenticated | 4901170000000003 | |
| Frictionless | - | - | Authenticated | 4901164281364345 | |
| Frictionless | - | - | Authenticated | 4378510000000004 | |
| Frictionless | - | - | Attempted | 4456530000001005 | |
| Frictionless | - | - | Attempted | 5200000000001054 | |
| Frictionless | Y | - | Authenticated | 4066330000000004 | |
| Frictionless | Y | - | Authenticated | 5200000000001021 | |
| Low risk exemption accepted (MasterCard) | - | - | Authenticated | 5169750000001111 | Used only for synchronous 3DS workflow. |
| Low risk exemption accepted (Visa) | - | - | Authenticated | 4378510000000004 | Used only for synchronous 3DS workflow. |
| Frictionless | - | - | Not authenticated | 4111110000000922 | |
| Frictionless | - | - | Not authenticated | 5200000000001047 | |
| Frictionless | Y | - | Not authenticated | 4111112232423922 | |
| Challenge | - | Y | Choose Challenge result | 4918190000000002 | |
| Challenge | - | Y | Choose Challenge result | 5433300000000133 | |
| Challenge | Y | Y | Choose Challenge result | 4938730000000001 | |
| Challenge | Y | Y | Choose Challenge result | 5200000000001005 | |

# Authentication Services

## Introduction

Authentication Services provide Strong Customer Authentication (SCA) which is a type of authentication relying on two or more independent elements.

SCA as defined by the European Central Bank (ECB) and in the context of the EU’s Payment Services Directive (PSD2) is:

”a procedure based on the use of two or more of the following elements categorised as knowledge, ownership and inherence:

- something only the user knows, e.g. static password, code, personal identification number;
- something only the user possesses, e.g.token, smart card, mobile phone;
- something the user is, e.g. biometric characteristic, such as a fingerprint.

In addition, the elements selected must be mutually independent, i.e. the breach of one does not compromise the other(s).

At least one of the elements should be non-reusable and non-replicable (except for inherence), and not capable of being surreptitiously stolen via the Internet.

The strong authentication procedure should be designed in such a way as to protect the confidentiality of the authentication data.”

Using multiple solutions from the same category would not constitute SCA.

**SCA WITH 3DSECURE**

Only the newest variant of 3DSecure involving one-time passwords (OTP) constitutes a form of SCA.

3DSecure however might be a weak SCA solution for several reasons:

1. it relies on the cards being actively enrolled by the cardholder after issuing, i.e prior any transaction made with it and
2. it is not available on all card schemes.

**ALTERNATIVE SERVICES**

A number of services exist that provide SCA in a compliant way.

They rely on different solutions like generating payment ”secrets”, one-time passwords etc. Key advantage of such services is that they allow cards to be enrolled ”on-the-fly”, i.e. during a transaction rather than having to be pre-enrolled after issuing.

Such services could potentially work with any card and are thus card scheme agnostic.

> Authentication services are not available by default, they need to be enabled on your account. Please contact Tech Support for further assistance.

# Genesis KYC Services

## General Info

Genesis KYC Services gives us the ability to perform particular checks on the integrity of the consumer data. Based on the returned consumer score we can decide whether we want to reject/approve a given transaction or perform another action for this consumer.

> You must contact support in order to obtain KYC service credentials.

## Create Consumer Registration

Review all aspects of the customer’s information, as it is received in the registration process, against local and external databases to increase accuracy and produce a risk score for that customer.

`POST /api/v1/create_consumer`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/create_consumer \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "customer_information": {
     "first_name": "John",
     "last_name": "Doe",
     "customer_email": "jdoe@example.com",
     "address1": "Sample Address",
     "address2": "4th floor",
     "city": "MyCity",
     "zip_code": "32132",
     "country": "BG",
     "province": "ProvinceName",
     "phone1": "0883113332",
     "phone2": "0883113334",
     "birth_date": "1987-03-03",
     "document_number": "f2345838972",
     "document_type": 3,
     "gender": "M"
   },
   "customer_unique_id": "21343253",
   "customer_registration_date": "2016-12-12",
   "customer_registration_ip_address": "255.255.255.255",
   "customer_username": "dimo16",
   "customer_registration_device_id": "12343242",
   "third_party_device_id": "3432424",
   "profile_action_type": 1,
   "device_fingerprint_type": 1,
   "current_profile_status": 1,
   "bonus_code": "1922",
   "bonus_amount": 100,
   "merchant_website": "dai.com",
   "how_did_you_hear": "friend",
   "affiliate_id": "1922"
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| session_id | optional | string | If this value is not provided the user email account should be complete and valid |
| customer_information | required | object | Customer information. See below for each of the nested required fields |
| customer_username | optional | string | Username of the customer on your system |
| customer_unique_id | required | string | Unique user identificator on your system |
| customer_registration_date | required | string | Date in which the customer was registered in the system OR the date in which the customer was created in the cashier Database yyyy-mm-dd |
| customer_registration_ip_address | required | string | IP address of customer used when the customer was registered in the system OR the current IP address |
| customer_registration_device_id | optional | string | Propietary DeviceId technology, refer to the DeviceId Instruction Manual (provided on request) |
| third_party_device_id | optional | string | Third Party DeviceId |
| device_fingerprint | optional | string | Open Source DeviceId technologies (Interpreted as a String) |
| device_fingerprint_type | optional | enum | 1 - Custom; 2 - Open Source; 3 - Open Source 2; |
| profile_action_type | optional | enum | 1 - Registration;2 - Profile Update; |
| profile_current_status | optional | enum | 0 - Undefined;1 - Review;2 - Denied;3 - Approved; |
| bonus_code | optional | string | Open text variable. Represents the code entered by the customer |
| bonus_submission_date | optional | string | |
| bonus_amount | optional | number | |
| merchant_website | optional | string | Website URL of the merchant initiating the wallet transfer |
| industry_type | optional | string | 1 - Finance; 2 - Gambling; 3 - Crypto; 4 - Travel; 5 - Retail; 6 - Risk Vendor; 7 - Adult; 8 - Remittance/Transfer; 9 - Other; |

| | | | |
|---|---|---|---|
| how_did_you_hear | optional | string | |
| affiliate_id | optional | string | |
| rule_context | optional | number | Number assigned to a given rule context. Please contact to get the available contexts |

`required* = conditionally required`

**Customer Information Fields**

The fields of the customer information object.

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| first_name | required | string | Customer first name |
| middle_name | optional | string | |
| last_name | required | string | Customer last name |
| customer_email | required | string | Must contain valid e-mail of customer |
| address1 | required | string | Primary address |
| address2 | optional | string | Secondary address |
| city | required | string | City |
| province | required | string | |
| zip_code | required | string | ZIP code |
| country | required | string | two-letter iso codes |
| phone1 | optional | number | |
| phone2 | optional | number | |
| birth_date | optional | string | Required for Visa only when MCC is a Financial Services one (e.g. MCC 6012) |
| document_type | optional | enum | 0 - SSN; 1 - Passport Registry; 2 - Personal ID / National ID; 3 - Identity Card; 4 - Driver License; 8 - Travel Document; 12 - Residence Permit; 13 - Identity Certificate; 16 - Registro Federal de Contribuyentes; 17 - Credencial de Elector; 18 - CPF |
| document_number | optional | string | |
| gender | optional | enum | F - female; M - male |

`required* = conditionally required`

Successful Response


```
{
  "code": 0,
  "message": "Successful Response",
  "technical_message": "Successful Response",
  "details": [
    {
      "reference_id": "233",
      "risk_score": 98,
      "kyc_provider_recommendation": "Reject",
      "rules_triggered": [
        {
          "name": "Multi-Accounting : IP shared to Chargeback reason",
          "risk_score": "100.00",
          "display_to_merchant": 1
        }
      ],
      "scrubber_results": {
        "geo_check": "",
        "address_verification": "",
        "phone_verify": "",
        "idv_usa": "",
        "idv_global": "",
        "gav": "",
        "idv_br": "",
        "bav_usa": "",
        "bav_advanced": "",
        "cb_aml": "",
        "cb_bvs": "",
        "email_age": "",
        "compliance_watchlist": "",
        "iovation": "",
        "idv_advance": ""
      },
      "result_confidence_level": 91.5
    }
  ]
}
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | genesis success code - 0 |
| message | string | const ‘Successful Response’ |
| technical_message | string | const ‘Successful Response’ |
| **details** | | |
| reference_id | string | KYC provider internal id |
| risk_score | number | risk score low is better |
| kyc_provider_recommendation | string | KYC service provider recommendation |
| result_confidence_level | number | Confidence level in percentages |
| ... | string | additional response data |

---

# Update Consumer Registration

Update the customer registration to be able to pass on the latest status required so we can continue improving the data models and provide the best scores and recommendations possible.

`POST /api/v1/update_consumer`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/update_consumer \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "reference_id": "9999333344443",
   "profile_current_status": 2,
   "status_reason": "Reject"
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| reference_id | required | number | Unique id returned by corresponding transaction |
| profile_current_status | required | enum | 0 - Undefined; 1 - Review; 2 - Denied; 3 - Approved; |
| status_reason | optional | string | Required only if status is Reject / Decline / Chargeback / Refund / Return / Void |

`required* = conditionally required`

Make sure that `reference_id` points to a preliminary created transaction (will describe transactions below).

Successful Response

```
 {
   "code": 0,
   "message": "Successful Response",
   "technical_message": "Successful Response",
   "details": [
     {
       "reference_id": "9999333344443"
     }
   ]
 }
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | genesis success code - 0 |
| message | string | const 'Successful Response' |
| technical_message | string | const 'Successful Response' |
| **details** | | |
| reference_id | string | KYC provider internal id |

---

# Create Transaction

Implement this to scrub a new transaction. We will take the information specific to that transaction and run various verification checks available, returning the recommendation, score, and third-party verification scrubbing results.

`POST /api/v1/create_transaction`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/create_transaction \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "customer_information": {
     "first_name": "John",
     "last_name": "Doe",
     "customer_email": "jdoe@example.com",
     "address1": "Sample address",
     "city": "Paris",
     "zip_code": "666",
     "country": "FR",
     "province": "MyProvince"
   },
   "deposit_limits": {
     "payment_method": "CC"
   },
   "transaction_unique_id": "1332",
   "payment_details": {
     "bin": "411111",
     "tail": "1111"
   },
   "customer_ip_address": "255.255.255.255",
   "transaction_created_at": "2016-12-12 23:23:23",
   "currency": "USD",
   "amount": 99999
 }'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/create_transaction \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "customer_information": {
     "first_name": "John",
     "last_name": "Doe",
     "customer_email": "jdoe@example.com",
     "address1": "Sample address",
     "city": "Paris",
     "zip_code": "666",
     "country": "FR",
     "province": "MyProvince"
   },
   "deposit_limits": {
     "payment_method": "CC"
   },
   "transaction_unique_id": "1332",
   "payment_details": {
     "ewallet_id": "test@example.com"
```

```
    },
    "customer_ip_address": "255.255.255.255",
    "transaction_created_at": "2016-12-12 23:23:23",
    "currency": "USD",
    "amount": 99999
  }'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/create_transaction \
   -X POST \
   -H "Content-Type: application/json" \
   -d '
  {
    "customer_information": {
      "first_name": "John",
      "last_name": "Doe",
      "customer_email": "jdoe@example.com",
      "address1": "Sample address",
      "city": "Paris",
      "zip_code": "666",
      "country": "FR",
      "province": "MyProvince"
    },
    "deposit_limits": {
      "payment_method": "CC"
    },
    "transaction_unique_id": "1332",
    "payment_details": {
      "bin": "411111",
      "tail": "1111",
      "cvv_present": "YES"
    },
    "customer_ip_address": "255.255.255.255",
    "transaction_created_at": "2016-12-12 23:23:23",
    "currency": "USD",
    "amount": 99999
  }'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/create_transaction \
   -X POST \
   -H "Content-Type: application/json" \
   -d '
  {
    "customer_information": {
      "first_name": "John",
      "last_name": "Doe",
      "customer_email": "jdoe@example.com",
      "address1": "Sample address",
      "city": "Paris",
      "zip_code": "666",
      "country": "FR",
      "province": "MyProvince"
    },
    "deposit_limits": {
      "payment_method": "EC"
    },
    "transaction_unique_id": "1332",
    "payment_details": {
      "routing": "1912",
      "account": "000001234512345123450000 0"
    },
    "customer_ip_address": "255.255.255.255",
    "transaction_created_at": "2016-12-12 23:23:23",
    "currency": "USD",
    "amount": 99999
  }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| session_id | optional | string | If this value is not provided the user email account should be complete and valid |
| customer_username | optional | string | Username of the customer on your system |
| customer_unique_id | optional | string | Unique user identificator on your system |
| customer_status | optional | string | Current status of the customer account in your system |
| customer_loyalty_level | optional | string | Customer loyalty level; for example: VIP; Bronze; Platinum; Gold; etc. This is an open text variable |
| customer_registration_date | optional | string | Date in which the customer was registered in the system OR the date in which the customer was created in the cashier Database yyyy-mm-dd |
| customer_registration_ip_address | optional | string | IP address of customer used when the customer was registered in the system OR the current IP address |
| customer_registration_device_id | optional | string | Propietary DeviceId technology, refer to the DeviceId Instruction Manual (provided on request) |
| customer_information | required | object | Customer information. See below for each of the nested required fields |
| first_deposit_date | optional | string | Empty if first deposit yyyy-mm-dd |
| first_withdrawal_date | optional | string | Empty if 0 withdrawals yyyy-mm-dd |
| deposits_count | optional | number | |
| withdrawals_count | optional | number | |
| current_balance | optional | number | |
| deposit_limits | required | object | See below |
| transaction_unique_id | required | string | Transaction id |
| billing_information | optional | object | See below |
| shipping_information | optional | object | See below |
| payment_details | required | object | See below |
| amount | optional | number | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | optional | string | ISO 4217 Three digits |
| transaction_created_at | required | string | Represents the time of the transaction on the Merchant server. Format: yyyy-mmdd hh:mm:ss |
| transaction_status | optional | enum | Transaction status; it is recommended to send 0 on the initial call. Afterwards call Update Transaction endpoint to update the status. 0 - numberUndefined; 1 - number- Approved; 2 - number- Pre-Auth; 3 - number- Settled; 4 - number- Void; 5 - number- Rejected internally by Negative Database or other scrubber decided to reject the transaction; 6 - number- Declined the bank / gateway / processor rejected the transaction; 7 - number- Chargeback; 8 - number- Return;9 - number- |

| | | | |
|---|---|---|---|
| | | | Pending; 10 - number- Pass Transaction validation;11 - number- Failed Transaction validation; 12 - number- efund; 13 - number- Approved Review; 14 - number- Abandon This status is used when the user just leaves the transaction; |
| customer_ip_address | required | string | Customers IP address |
| customer_device_id | optional | string | Propietary DeviceId technology; refer to the DeviceId Instruction Manual (provided on request) |
| third_party_device_id | optional | string | Third Party DeviceId |
| device_fingerprint | optional | string | Open Source DeviceId technologies (Interpreted as a String) |
| device_fingerprint_type | optional | enum | 1 - Custom; 2 - Open Source; 3 - Open Source 2; |
| shopping_cart_items_count | optional | number | Represents the quantity of items in the shopping cart |
| local_time | optional | string | Represents the local time of the customer doing the transaction. Format: yyyy-mmdd hh:mm:ss |
| order_source | optional | enum | internet; mobile; inhouse |
| merchant_website | optional | string | Open text variable; it represents the website name or URL that submitted the transaction |
| industry_type | optional | enum | Definition of the industry type the transaction was performed on; 1-number - Finance; 2-number - Gambling; 3-number - Crypto; -number - Travel; 5-number - Retail; 6-number - Risk Vendor; 7-number - Adult; 8-number - Remittance/Transfer; 9-number - Other; |
| customer_password | optional | string | Open text variable; it represents the customers password in hashed format (using MD5) some companies share that information in order to look for patterns |
| rule_context | optional | number | Number assigned to a given rule context. Please contact to get the available contexts. |
| custom_variable | optional | string | Represents anything the merchant wants to store with this transaction |

`required* = conditionally required`

**Deposit Limits Fields**

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| payment_method | required | enum | CC; EC - CreditCard; Echeck |
| minimum | optional | number | Lowest valid amount for deposit in minor currency units; ex: 100 = $1 |
| daily_maximum | optional | number | In minor currency units |
| weekly_maximum | optional | string | In minor currency units |
| monthly_maximum | optional | string | In minor currency units |

`required* = conditionally required`

**Billing Information Fields**

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| first_name | optional | string | Customer first name |
| last_name | optional | string | Customer last name |
| customer_email | optional | string | Must contain valid e-mail of customer |
| address1 | optional | string | Primary address |
| address2 | optional | string | Secondary address |
| city | optional | string | City |
| province | optional | string | |
| zip_code | optional | string | ZIP code |
| country | optional | string | ISO 3166-1 Alpha-2. For example: USD |
| phone1 | optional | number | Numbers only; no dash or any other separator. Please include area code if applicable. Country code is not required |
| birth_date | optional | string | yyyy-mm-dd |
| gender | optional | enum | M; F |

`required* = conditionally required`

**Shipping Information Fields**

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| first_name | optional | string | Customer first name |
| last_name | optional | string | Customer last name |
| customer_email | optional | string | Must contain valid e-mail of customer |
| address1 | optional | string | Primary address |
| address2 | optional | string | Secondary address |
| city | optional | string | City |
| province | optional | string | |
| zip_code | optional | string | ZIP code |
| country | optional | string | ISO 3166-1 Alpha-2. For example: USD |
| phone1 | optional | number | Numbers only; no dash or any other separator. Please include area code if applicable. Country code is not required |

`required* = conditionally required`

**Payment Details Fields**

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| bin | optional | string | First 6 digits of the card number; Only required if Payment Method Type is Credit Card; |

| | | | |
|---|---|---|---|
| tail | optional | string | Last 4 digits of the card number; Only required if Payment Method Type is Credit Card; |
| cvv_present | optional | string | Indicator if the CVV was received or not; The expected values in this field are Yes or No; |
| hashed_pan | optional | string | Only required if Payment Method Type is Credit Card; It should be hashed using SHA256; the string to be hashed is Card Number and the MD5 hash of the Expiration Date; For example; if the card number is 1111-2222-3333-4444 with expiration date 01/22; The have should be done on the string without spaces nor dash nor other special chars; The MD5 of the Expiration Date 0122 is f0f4b6598f2cee45644673998b4f44be; That said; the string to be hashed is 1111222233334444f0f4b6598f2cee45644673998b4f44be which generates the following result fecd244b7d647b0db391e35910e0d42aaf88f7633a4f4f4883b109abad1d6d7 |
| routing | optional | string | Routing number; Only required if Payment Method Type is eCheck; |
| account | optional | string | Only numbers up to 30 digits; Only required if Payment Method Type is eCheck; |
| ewallet_id | optional | string | Most of the times its an email; Only required if Payment Method Type is eWallet; |

`required* = conditionally required`

Required attributes, depending on the Payment Method:

- For payment method CC
  - bin
  - tail
- For payment method CC OPTIONAL
  - cvv_present
  - hashed_pan
- For payment method EC
  - routing
  - account

Successful Response


```
{
  "code": 0,
  "message": "Successful Response",
  "technical_message": "Successful Response",
  "details": [
    {
      "reference_id": "481119",
      "kyc_provider_recommendation": "Approve",
      "risk_score": 0.39,
      "third_party": "",
      "processors": "",
      "reason": "",
      "result_confidence_level": 91.5,
      "rules_triggered": [
        {
          "name": "Address Verification : Global 25",
          "score": "0",
          "display_to_merchant": 1
        }
      ],
      "bin_information": {
        "bank_name": "BANCO NACIONAL DE COSTA RICA",
        "bank_location": "COSTA RICA",
        "card_type": "STANDARD",
        "card_level": "DEBIT",
        "iso_card_country": "CR",
        "card_brand_db": "MASTERCARD",
        "card_brand_script": "MASTERCARD"
      },
      "scrubber_results": {
        "geo_check": "",
        "address_verification": "",
        "phone_verify": "",
        "idv_usa": "",
        "idv_global": "",
        "gav": "",
        "idv_br": "",
        "bav_usa": "",
        "bav_advanced": "",
        "cb_aml": "",
        "cb_bvs": "",
        "email_age": "",
        "compliance_watchlist": "",
        "iovation": "",
        "idv_advance": ""
      }
    }
  ]
}
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | genesis success code - 0 |
| message | string | const ‘Successful Response’ |
| technical_message | string | const ‘Successful Response’ |
| **details** | | |
| reference_id | string | KYC provider internal id |
| kyc_provider_recommendation | string | KYC provider recommendation |
| risk_score | number | risk score low is better |
| result_confidence_level | number | KYC service level of decision confidence |
| ... | * | KYC provider additional data |

---

# Update Transaction

Utilize this method to update a particular transaction status so we can continue improving the data models and provide the best scores and recommendations.

`POST /api/v1/update_transaction`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/update_transaction \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
  {
```

```
    "transaction_status": 1,
    "reference_id": "4454982",
    "reason": "a reason message",
    "transaction_unique_id": "1332",
    "cvv_check_result": "",
    "avs_check_result": "",
    "processor_identifier": ""
  }'
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/update_transaction \
   -X POST \
   -H "Content-Type: application/json" \
   -d '
  {
    "transaction_status": 2,
    "transaction_unique_id": "1332"
  }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| session_id | optional | string | If this value is not provided the user email account should be complete and valid |
| transaction_unique_id | required | string | Transaction id |
| reference_id | optional | number | Required only if status is Reject / Decline / Chargeback / Refund / Return / Void |
| transaction_status | optional | enum (number) | 1-Approved; 2-Pre-Auth; 3-Settled; 4-Void; 5-Rejected internally by Negative Database or other scrubber decided to reject the transaction; 6-Declined the bank / gateway / processor rejected the transaction; 7-Chargeback; 8-Return; 9-Pending; 10-Pass Transaction validation; 11-Failed transaction validation; 12-Refund; 13-Approved Review; 14-Abandon This status is used when the user just leaves the transaction; |
| reason | optional | string | Required only if status is Reject / Decline / Chargeback / Refund / Return / Void |
| cvv_check_result | optional | string | Response from processor regarding CVV check |
| avs_check_result | optional | string | Result from processor regarding AVS check |
| processor_identifier | optional | string | Unique identifier of the processor attempted |

`required* = conditionally required`

Successful Response

```
  {
    "code": 0,
    "message": "Successful Response",
    "technical_message": "Successful Response",
    "details": [
      {
        "reference_id": "481119"
      }
    ]
  }
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | genesis success code - 0 |
| message | string | const ‘Successful Response’ |
| technical_message | string | const ‘Successful Response’ |
| **details** | | |
| reference_id | string | KYC provider internal id |

---

# Identity Document Upload

Used to verify documents provided by the customer.

When called this action returns the following:

- An instant response with a reference id and additional keys by which the particular query could be addressed.
- Async responses with answers from the performed checks (OCR, manual, etc.). Usually, this responses arrive after 5-7 minutes. The notification url is agreed on environment basis. Create a new transaction for the particular user before making a call of this kind.

`POST /api/v1/upload_document`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/upload_document \
   -X POST \
   -H "Content-Type: application/json" \
   -d '
  {
    "transaction_unique_id": "1332",
    "doc": {
      "mime_type": "image/jpeg",
      "base64_content": "{base_64_endcoded_content}"
    }
  }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| customer_username | optional | string | Username of the customer on your system |
| customer_unique_id | optional | string | Unique user identificator on your system |
| transaction_unique_id | required* | string | Unique Transaction Id with info of the customer to be verified. Please note; if Transaction Id and Customer Registration Id are provided the system will use the Transaction Id. Please provide the Transaction Id or the Customer Registration Id; one of them must be provided |
| reference_id | required* | string | Unique Customer Registration Id with info of the customer to be verified |
| method | required | enum (number) | 1 - Manual; 2 - OCR; 3 - Both; |
| doc | required | object | see below |

| | | | |
|---|---|---|---|
| doc2; doc3; doc4 | optional | object | additional document images |

`required* = conditionally required`

One of `transaction_unique_id` and `reference_id` fields is required.

**Document Fields**

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| base64_content | required | string | |
| mime_type | required | string | |

`required* = conditionally required`

Successful Response

```
{
  "code": 0,
  "message": "Successful Response",
  "technical_message": "Successful Response",
  "details": [
    {
      "kyc_source": "OCR",
      "reference_id": "382"
    },
    {
      "kyc_source": "Semi-manual",
      "reference_id": "129"
    }
  ],
  "doc": "51f7e411cd1202e040b21655738beb89",
  "doc2": "b0139d3fb1785c1e57ac5a6b054692eb",
  "doc3": "e0059153192876ce8ab36d8cd8ab1bca"
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | genesis success code - 0 |
| message | string | const 'Successful Response' |
| technical_message | string | const 'Successful Response' |
| **details** | | |
| kyc_source | string | KYC service name triggered for verification |
| reference_id | string | id used for reference when verification result is received |

---

# Identity Document Download

Uploaded documents will be stored by legal provisions and they can be requested for review. Just post a JSON body with the identity document id of the given document and a response with the filename and the base64 encoded content of that file would be returned.

`POST /api/v1/download_document`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/download_document \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
  {
    "identity_document_id": "676a053b16781e43db7e75dcb1444ef8"
  }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| identity_document_id | required | string | document id |

`required* = conditionally required`

Successful Response

```
{
  "code": 0,
  "message": "Successful Response",
  "technical_message": "Successful Response",
  "details": [
    {
      "document": {
        "file_name": "430d595c22dbae26ef39ed91c3aabb49.jpg",
        "base64_content": "{base_64_endcoded_content}"
      }
    }
  ]
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | genesis success code - 0 |
| message | string | const 'Successful Response' |
| technical_message | string | const 'Successful Response' |
| **details** | | |
| file_name | string | file name |
| base64_content | string | base64 encoded document content |

# Make call

This method is used to make a call or send an SMS to a given phone number. This method is used to complement the verification process. The system will make a call and dictates the verification code to be typed in the website. The following is a transcript of the voice message the system will use when en-US is used as language:

> Hello, thank you for using our phone verification system. Your code is [CODE]. Once again, your code is [CODE]. Goodbye!

The following is an example of the SMS text the system will use when en-US is used as language:

> Your code is [CODE]. Thank you.

`POST /api/v1/create_authentication`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/create_authentication \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "customer_username": "896342",
   "customer_unique_id": "2",
   "transaction_unique_id": "387783428790324",
   "customer_phone_number": "372489879342",
   "service_language": "bg",
   "security_code": "3423",
   "service_type": 1
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| customer_username | optional | string | Username of the customer on your system |
| customer_unique_id | optional | string | Unique user identificator on your system |
| transaction_unique_id | required | string | Transaction identification in the merchants system; If not provided the system won't be able to link with the transaction that is being verified |
| customer_phone_number | required | string | Phone number to call; It must be complete country code + phone number; No dashes; For example: 50622560000 |
| service_language | required | string | a string value; See below |
| security_code | required | string | Numeric value - 4 digits only; It cannot start with 0; The boot is going to say this numeric value so the user can type it back on the website; |
| service_type | required | string | Numeric value to indicate if the system will send a text message or make a voice call; 1 for SMS; 2 for Voice call; |

`required* = conditionally required`

**Available languages for t_language**

| Language | Code |
|---|---|
| Arabic | a |
| Cantonese, Chinese/Hong Kong | zh-HK |
| Catalan | ca |
| Croatian | hr |
| Czech | cs |
| Danish | da |
| Dutch | nl |
| English, Australian | en-AU |
| English, UK | en-GB |
| English, US | en-US |
| Estonian | et |
| Filipino | fil |
| Finnish | fi |
| French | fr |
| French, Canadian | fr-CA |
| German | de |
| Greek | el |
| Hebrew | he |
| Hindi | hi |
| Hungarian | hu |
| Icelandic | is |
| Indonesian | id |
| Italian | it |
| Japanese | ja |
| Korean | ko |
| Latvian | lv |
| Lingala | ln |
| Lithuanian | lt |
| Mandarin | zh-CN |
| Norwegian | no |

| | |
|---|---|
| Polish | pl |
| Portuguese, Brazilian | pt-BR |
| Portuguese, European | pt |
| Romanian | ro |
| Russian | ru |
| Slovakian | sk |
| Spanish, European | es |
| Spanish, Latin American | es-419 |
| Swedish | sv |
| Thai | th |
| Turkish | tr |
| Ukrainian | uk |
| Vietnamese | vi |

Successful Response

```
{
  "code": 0,
  "message": "Successful Response",
  "technical_message": "Successful Response",
  "details": [
    {
      "reference_id": "233"
    }
  ]
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | genesis success code - 0 |
| message | string | const 'Successful Response' |
| technical_message | string | const 'Successful Response' |
| **details** | | |
| reference_id | string | id used for future reference |

---

# Update call

This method is used to update the call status with the latest info received from the main system. It also updates the transaction associated with this verification call.

`POST /api/v1/update_authentication`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/update_authentication \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
{
  "reference_id": 1237834,
  "security_code_input": "4322",
  "verification_status": "4"
}'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| reference_id | required | string | Unique value to identify the call in back office. |
| security_code_input | required | string | Transaction identification in the merchants system; If not provided the system won't be able to link with the transaction that is being verified |
| verification_status | required | string | The first two values are defined by the system when the call is created; the ones accepted in this call are the status 3; 4 and 5 only 1-In Progress; 2-Failed; 3-Verification Failed; 4-Verification Successful; 5-Abandon; |

`required* = conditionally required`

Successful Response

```
{
  "code": 0,
  "message": "Successful Response",
  "technical_message": "Successful Response",
  "details": [
    {
      "reference_id": "233"
    }
  ]
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | genesis success code - 0 |
| message | string | const 'Successful Response' |
| technical_message | string | const 'Successful Response' |
| **details** | | |
| reference_id | string | id used for future reference |

# Create Verification

The verification request will provide a link that will be used to redirect the customer. The customer will provide the required documents and will be verified against them. As a result, the user will be redirected back to merchant based on the provided redirect URL.

`POST /api/v1/verifications`

> Be aware that if neither `document_supported_types` nor `redirect_url` parameters are passed, the verification will be performed synchronously and either `accepted` or `declined` status will be returned.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/verifications \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "email": "test@email.com",
   "country": "US",
   "language": "EN",
   "redirect_url": "https://merchant.com/from_kyc",
   "reference_id": "123456",
   "document_supported_types": [
     "id_card",
     "passport"
   ],
   "address_supported_types": [
     "id_card",
     "passport"
   ],
   "face": {
     "allow_offline": true,
     "allow_online": true,
     "check_duplicate_request": true
   },
   "backside_proof_required": true,
   "address_backside_proof_required": true,
   "expiry_date": "2021-10-10",
   "background_checks": {
     "date_of_birth": "1995-10-10",
     "async_update": true,
     "first_name": "John",
     "middle_name": "Carter",
     "last_name": "Doe",
     "full_name": "John Carter Doe",
     "filters": [
       "sanction",
       "warning",
       "fitness-probity",
       "pep",
       "pep-class-1",
       "pep-class-2",
       "pep-class-3",
       "pep-class-4",
       "adverse-media"
     ],
     "match_score": 100
   },
   "document": {
     "date_of_birth": "1995-10-10",
     "first_name": "John",
     "last_name": "Doe",
     "allow_offline": false,
     "allow_online": true
   },
   "allow_retry": true,
   "verification_mode": "image_only"
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| email | required | string | User's email |
| country | optional | string | Country code in ISO 3166 |
| language | optional | string | Supported Language Code. Check Supported Languages |
| redirect_url | optional | string | URL where the customer is sent to after completing the verification process |
| reference_id | optional | string | Unique value to identify the performed verification. Should be not less than 6 characters, and not more than 250 characters. |
| document_supported_types | optional | array | Supported types of document that can be verified. Check Supported Document Types |
| address_supported_types | optional | string | Supported types of address that can be verified, Check Supported Address Types |
| **face** | | | Represents the options to be supplied to the service in order to provide face verification functionality |
| allow_offline | optional | boolean | Whether uploading of previously taken picture is allowed for verification of document/face |
| allow_online | optional | boolean | Whether the realtime usage of device camera is allowed for verification of document/face |
| check_duplicate_request | optional | boolean | Whether to enable the duplicate account detection service |
| backside_proof_required | optional | boolean | Signifies that both sides of the document are required to verify the identity |
| address_backside_proof_required | optional | boolean | Signifies that both sides of the document are required to verify the address |
| expiry_date | optional | yyyy-mm-dd | Document's expiry date at yyyy-mm-dd format, for example - 2025-12-31, can be a blank string. A blank string means that the user will need to enter the expiry date from the UI |
| allow_retry | optional | boolean | If the parameter value is set to 'true', the customer will be able to retry if the verification request is declined by the AI. On retry, the customer can re-upload the verification proof after correcting the indicated issues. |
| verification_mode | optional | string | This key specifies the types of proof that can be used for verification. Check Supported Address Types |
| **background_checks** | optional | | An AML (anti-money laundering) background check will be done based on the provided data. Please note that the name and the date of birth keys will be extracted from the document service if they are missing. |
| first_name | optional | string | Customer's first name |
| middle_name | optional | string | Customer's middle name |
| last_name | optional | string | Customer's last name |
| full_name | optional | string | Customer's full name |
| date_of_birth | optional | yyyy- | Customer's date of birth. just, without at yyyy-mm-dd format, for example - 1990-12-31 |

| | | | |
|---|---|---|---|
| | | mm-dd | |
| async_update | optional | boolean | Will allow the system to send notifications with information about the checked person when the status has been changed. The registered asynchronous update doesn't expire and notification will be sent on each change, but not often than 15 minutes |
| filters | optional | array | One or more AML Filters to query. Check AML Filters for more information |
| match_score | optional | number | Minimum match score to consider the match successful. Default: 100 |
| **document** | optional | | Document represents the data used by the document verification service to check the authenticity of identity documents submitted by customers |
| first_name | optional | string | Customer's first name |
| last_name | optional | string | Customer's last name |
| date_of_birth | optional | yyyy-mm-dd | Customer's date of birth. just, without at yyyy-mm-dd format, for example - 1990-12-31 |
| allow_offline | optional | boolean | Whether uploading of previously taken picture is allowed for verification of document/face |
| allow_online | optional | boolean | Whether the realtime usage of device camera is allowed for verification of document/face |

`required* = conditionally required`

Successful Response

```
{
  "code": 0,
  "message": "Successful Response",
  "technical_message": "Successful Response",
  "status": "pending",
  "verification_url": "https://app.shuftipro.com/process/kyc/code",
  "reference_id": "123456"
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | 0 - for successful response or genesis error code, see below |
| message | string | code description, 'Successful Response' for code - 0 |
| technical_message | string | 'Successful Response', or technical error description |
| status | string | Status of verification, Check Available Statuses Types |
| verification_url | string | URL that should be used by end user to submit verification process |
| reference_id | string | Unique value to identify the performed verification |

**SUPPORTED DOCUMENT TYPES**

| Supported Types |
|---|
| passport |
| id_card |
| driving_license |
| credit_or_debit_card |

**SUPPORTED ADDRESS TYPES**

| Supported Types |
|---|
| id_card |
| passport |
| driving_license |
| utility_bill |
| bank_statement |
| rent_agreement |
| employer_letter |
| insurance_agreement |
| tax_bill |
| envelope |
| cpr_smart_card_reader_copy |

**AVAILABLE STATUSES**

| Status |
|---|
| pending |
| error |
| cancelled |
| timeout |
| error |
| accepted |
| declined |
| changed |
| deleted |
| received |

**SUPPORTED LANGUAGES**

| Country Name | Language Code |
|---|---|
| | |

| | |
|---|---|
| Afrikaans | AF |
| Albanian | SQ |
| Amharic | AM |
| Arabic | AR |
| Armenian | HY |
| Azerbaijani | AZ |
| Basque | EU |
| Belarusian | BE |
| Bengali | BN |
| Bosnian | BS |
| Bulgarian | BG |
| Burmese | MY |
| Catalan | CA |
| Chichewa | NY |
| Chinese | ZH |
| Corsican | CO |
| Croatian | HR |
| Czech | CS |
| Danish | DA |
| Dutch | NL |
| English | EN |
| Esperanto | EO |
| Estonian | ET |
| Filipino | TL |
| Finnish | FI |
| French | FR |
| Frisian | FY |
| Galician | GL |
| Georgian | KA |
| German | DE |
| Greek (modern) | EL |
| Gujarati | GU |
| Haitian, Haitian Creole | HT |
| Hausa | HA |
| Hebrew (modern) | HE |
| Hindi | HI |
| Hungarian | HU |
| Indonesian | ID |
| Irish | GA |
| Igbo | IG |
| Icelandic | IS |
| Italian | IT |
| Japanese | JA |
| Javanese | JV |
| Kannada | KN |
| Kazakh | KK |
| Khmer | KM |
| Kirghiz, Kyrgyz | KY |
| Korean | KO |
| Kurdish | KU |
| Latin | LA |
| Luxembourgish, Letzeburgesch | LB |
| Lao | LO |
| Lithuanian | LT |
| Latvian | LV |
| Macedonian | MK |
| Malagasy | MG |
| Malay | MS |
| Malayalam | ML |
| Maltese | MT |

| | |
|---|---|
| Maori | MI |
| Marathi | MR |
| Mongolian | MN |
| Nepali | NE |
| Norwegian | NO |
| Punjabi | PA |
| Persian | FA |
| Polish | PL |
| Pashto | PS |
| Portuguese | PT |
| Romanian | RO |
| Russian | RU |
| Sindhi | SD |
| Samoan | SM |
| Serbian | SR |
| Scottish Gaelic | GD |
| Shona | SN |
| Sinhala | SI |
| Slovak | SK |
| Slovenian | SL |
| Somali | SO |
| Sesotho | ST |
| Spanish | ES |
| Sundanese | SU |
| Swahili | SW |
| Swedish | SV |
| Tamil | TA |
| Telugu | TE |
| Tajik | TG |
| Thai | TH |
| Turkish | TR |
| Ukrainian | UK |
| Urdu | UR |
| Uzbek | UZ |
| Vietnamese | VI |
| Welsh | CY |
| Xhosa | XH |
| Yiddish | YI |
| Yoruba | YO |
| Zulu | ZU |

**SUPPORTED VERIFICATION MODES**

| Verification Modes |
|---|
| any |
| image_only |
| video_only |

**AML FILTERS**

| AML Filters | Default |
|---|---|
| sanction | Yes |
| warning | Yes |
| fitness-probity | Yes |
| pep | Yes |
| pep-class-1 | Yes |
| pep-class-2 | Yes |
| pep-class-3 | Yes |
| pep-class-4 | Yes |
| adverse-media | Yes |
| adverse-media-general | No |
| adverse-media-narcotics | No |
| adverse-media-fraud | No |

| | |
|---|---|
| adverse-media-terrorism | No |
| adverse-media-sexual-crime | No |
| adverse-media-violent-crime | No |
| adverse-media-financial-crime | No |
| adverse-media-v2-other-minor | No |
| adverse-media-v2-other-serious | No |
| adverse-media-v2-other-financial | No |
| adverse-media-v2-violence-non-aml-cft | No |
| adverse-media-v2-financial-difficulty | No |
| adverse-media-v2-regulatory | No |
| adverse-media-v2-general-aml-cft | No |
| adverse-media-v2-cybercrime | No |
| adverse-media-v2-terrorism | No |
| adverse-media-v2-violence-aml-cft | No |
| adverse-media-v2-narcotics-aml-cft | No |
| adverse-media-v2-fraud-linked | No |
| adverse-media-v2-financial-aml-cft | No |
| adverse-media-v2-property | No |

---

## Verification Status

Verification status check request can be performed by reference_id. A status check may be needed if async notification has not arrived yet or when kyc notifications are not used in general.

`POST /api/v1/verifications/status`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/verifications/status \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "reference_id": "2021-01-27-14:55:22-abb6966a-a6d4-4a3a-a364-ea3347384a90"
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| reference_id | required | string | Unique value to identify the performed verification |

`required* = conditionally required`

Successful Response

```
 {
   "code": 0,
   "message": "Successful Response",
   "technical_message": "Successful Response",
   "status": "timeout",
   "reference_id": "2021-01-27-14:55:22-abb6966a-a6d4-4a3a-a364-ea3347384a90",
   "proofs": {
     "document": {
       "proof": "https://api.shuftipro.com/storage/proof.png",
       "additional_proof": "https://api.shuftipro.com/storage/additionalproof.png"
     },
     "access_token": "6bf8a72b18062532576b30a2da8b881bbcb1bd4727a97a439270bec90943478e"
   },
   "background_checks": {
     "aml_filters": [
       "sanction",
       "pep",
       "pep-class-1"
     ],
     "aml_hits": [
       {
         "entity_type": "person",
         "score": 33.411358,
         "match_types": [
           "name_exact",
           "year_of_birth"
         ],
         "sources": [
           "european-union-council",
           "complyadvantage-adverse-media"
         ],
         "types": [
           "adverse-media",
           "pep",
           "pep-class-1"
         ],
         "name": "Boyko John Smith",
         "associates": [
           {
             "association": "spouse",
             "name": "Sue Smith"
           }
         ]
       }
     ]
   },
   "verification": {
     "document_expiry_date": "2030-11-12"
   }
 }
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| code | number | 0 - for successful response or genesis error code, see below |
| message | string | code description, ’Successful Response’ for code - 0 |
| technical_message | string | ’Successful Response’, or technical error description |
| status | string | Status of verification, Check Available Statuses Types |
| reference_id | string | Unique value to identify the performed verification |
| **proofs** | | |
| **document** | | |
| proof | string | Link to the proof document uploaded by the user (access_token is required to access it). |
| additional_proof | string | Link to any additional proof documents uploaded by the user (access_token is required to access it). |
| access_token | string | Token that must be included in POST requests to access documents from the proofs object. |
| **background_checks** | | |
| aml_filters | array | Applied AML filters |
| **aml_hits** | | |
| entity_type | string | Type of checked entry |
| score | float | Compliance score |
| sources | array | Sources of information |
| match_types | array | Contains AML match types |
| types | array | Contains AML types |
| name | string | Name of checked entry |
| associates | array | Information about the associates |
| **verification** | | |
| document_expiry_date | string | Expiration date of used identity document |

---

## Verification Register

Verification register request can be performed by reference_id. A reference id registration allows you to store the reference id in Genesis and receive notifications in Genesis for it.

`POST /api/v1/verifications/register`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/verifications/register \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "reference_id": "2021-01-27-14:55:22-abb6966a-a6d4-4a3a-a364-ea3347384a90"
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| reference_id | required | string | Unique value to identify the performed verification |

`required* = conditionally required`

Successful Response

```
{
  "code": 0,
  "message": "Successful Response",
  "technical_message": "Successful Response",
  "reference_id": "2021-01-27-14:55:22-abb6966a-a6d4-4a3a-a364-ea3347384a90"
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | 0 - for successful response or genesis error code, see below |
| message | string | code description, ’Successful Response’ for code - 0 |
| technical_message | string | ’Successful Response’, or technical error description |
| reference_id | string | Unique value to identify the performed verification |

---

## Error Response

Error Response

```
{
  "code": 404,
  "message": "Passed attribute is invalid!",
  "technical_message": "The property ‘#/’ did not contain a required property of ‘customer_information’"
}
```

Error Response

```
{
  "code": 803,
  "message": "KYC Services not configured!",
  "technical_message": "KYC Services not configured for Merchant!"
}
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| code | number | Genesis internal error code |
| message | string | Short explanation of occurred error |
| technical_message | string | More detailed explanation of occurred error |

---

# Kyc Service Notification

Since identity document verification is time consuming(can take up to 3-7 minutes) it is an asynchronous call. Notification params may vary based on the performed checks. When verification is done a notification will be sent to configured Merchant notification url with all the details of the review along with the risk score. Note that multiple notifications can be expected so please always check the Score Complete parameter as if we have multiple DocumentId Verify providers, the score will be complemented with more info as we receive the data from the KYC sources. The count of expected notifications will be declared in initial IdentityDocumentUpload response.

KYC Notification Example

```
?kyc_source=OCR
&reference_id=1912
&score=10
&score_complete=0
&uid=5486354658
&error_message=
&analysis_ref_uid=148871
&controls_identifier=MODEL_VALIDITY
&controls_title_msg=OK
&controls_result_msg=OK
&controls_result=OK
&controls_control_identifier=MODEL_RECOGNIZED
&controls_control_title_msg=OK
&controls_control_result_msg=OK
&controls_control_result=OK
&document_classification_id_type=V
&document_detail_emit_country=USA
&document_detail_expiration_date_day=23
&document_detail_expiration_date_month=12
&document_detail_expiration_date_year=2004
&document_detail_document_number=555123ABC
&document_detail_extra_infos_data_key=PERSONAL_NUMBER
&document_detail_extra_infos_data_value=IFLND00AMS803085
&document_detail_extra_infos_title=Personal_Number
&holder_detail_last_name=TRAVELER
&holder_detail_first_name=HAPPYPERSON
&holder_detail_nationality=GBR
&holder_detail_gender=F
&holder_detail_birth_date_day=5
&holder_detail_birth_date_month=2
&holder_detail_birth_date_year=1965
&mrz_line1=VIUSATRAVELER_HAPPYPERSON
&mrz_line2=555123ABC6GBR6502056F04122361FLND00AMS803085
&check_report_summary_check_identifier=SUMMARY_ID_COPY
&check_report_summary_check_title_msg=OK
&check_report_summary_check_result_msg=Original_Document
&check_report_summary_check_result=OK
&notification_type=kyc_service_execution
&signature=secure-signature
```

KYC Verification Request Notification Example

```
?reference_id=2021-01-28-09%3A47%3A50-443bc6b7-1f89-4774-a1f7-24eaf345d605
&verification_url=https%3A%2F%2Fapp.shuftipro.com%2Fprocess%2Fkyc%2FNl7Jpjbg2KIrGuWRUcAx5A7f11h8K1d4gkS1J8SUBaaIepuQ46izboRUEJDkTISq
&status=cancelled
&declined_reason
&email=john.doe1%40example.com
&notification_type=kyc_service_execution
&additional_data=additional_data_object
&signature=secure-signature
```

**Notification Parameters Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| kyc_source | string | name of the KYC source performed the validation |
| reference_id | string | Unique id for reference to document upload request |
| risk_score | number | Score product of validation rules |
| external_unique_id | string | Unique ID Reference from External Service Provider |
| analysis_reference_id | string | UniqueID Reference from Genesis |
| controls_identifier | string | List of all MAIN controls performed on the document |
| ... | string | Additional information related to verification process |
| notification_type | string | constant value ”kyc service response” |
| status | string | Status of verification, Check Available Statuses Types |
| verification_url | string | URL that should be used by end user to submit verification process, receives only for ”pending” status |
| email | string | User's email |
| declined_reason | string | Declined verification reason |
| additional_data | object | Additional data information related to verification request notification |
| signature | string | the signature of the notification, should be used to verify the the notification was sent by Genesis |

The signature is a security measure meant to ensure that the gateway is really the sender of the notification. It is generated by concatenating the reference id with your API login and generating a SHA-512 Hash (Hex) of the string:

`SHA-512 Hash Hex of [reference_id][Your Merchant API login]`

**Notification signature examples**

| reference_id | API login | signature |
|---|---|---|
| 1912 | login1 | 38b4f52584e6b1393db1503ee1ac10d10af2b39b69bfdff9828baebcec33e430e08c0013e5f3309ad5363458523084f81f21a5aad216c60933a470c9f08b8aca |
| 1818 | login2 | 0e40af69db1c06832fa66bc37bdd79c7a4b795810d92e50b2daaa6b8a8db9ff83524ccc3b3afbd5cb4dc99126042acb16bdf8060117acb09cad326cbe34279ee |

When receiving the notification, you are required to render an XML page containing the transaction’s reference id so that the gateway knows that you have accepted the notification. If the XML is not delivered, the notification

is sent periodically as per the rules for notifications delivery.

Notification Response Response

```
<?xml version="1.0" encoding="UTF-8"?>
<notification_echo>
  <notification_echo>reference_id</notification_echo>
</notification_echo>
```

---

# CPF check

Check the status of a specific Brazilian personal's CPF number. The CPF number is provided in the request URL as a parameter.

`POST /api/v1/cpf/DOCUMENT_ID`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/cpf/40442820135 \
  -X POST \
```

Successful Response

```
{
  "document_id": "40442820135",
  "account_holder": "Nome do CPF 404.428.201-35",
  "status": {
    "code": "active",
    "description": "Active"
  },
  "date_of_birth": "14111970"
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| document_id | string | individual taxpayer identification number |
| account_holder | string | name of Individual |
| **status** | | |
| code | string | registration status of the individual taxpayer identification number |
| description | string | description of the registration status of the individual taxpayer identification number |
| date_of_birth | string | birthdate of individual, in the format DDMMYYYY |

---

# CNPJ check

Check the status of a specific Brazilian company's CNPJ number. The CNPJ number is provided in the request URL as a parameter.

`POST /api/v1/cnpj/DOCUMENT_ID`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.kyc.e-comprocessing.net/api/v1/cnpj/54447820000155 \
  -X POST \
```

Successful Response

```
{
  "document_id": "54447820000155",
  "establishment_type": "Parent company",
  "company_name": "OWMKX NX OIWUQD UW",
  "fancy_name": "NQIHKWX BHIWD",
  "register_situation": {
    "code": "active",
    "description": "Active",
    "situation_date": "2005-11-03"
  },
  "legal_nature": {
    "code": "2038",
    "description": "GovernmentControlled Company",
    "open_date": "1966-08-01"
  }
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| document_id | string | individual taxpayer identification number |
| establishment_type | string | definition of the company, parent company or subsidiary |
| company_name | string | company name of the legal entity |
| fancy_name | string | legal name of legal Entity |
| **register_situation** | | |
| code | string | registration status code |
| description | string | description of the cadastral situation |
| situation_date | string | date of the cadastral situation |
| **legal_nature** | | |
| code | string | legal nature code |
| description | string | description of legal nature |
| open_date | string | establishment date |

---

# Genesis Fx Services

## General Info

Genesis Fx(Forex) Services provides the ability to retrieve up-to-date Fx rates. The API is synchronous and is based on RESTFul practices. Be sure to set Content-type: application/json in your headers.

To interact with the Fx API, you need to provide login credentials using standard HTTP Basic Authentication. (credentials can be found in your Admin interface.)

## Get Tiers

This call is used to return all Tiers that are related to your account.

`GET /v1/fx/tiers`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/fx/tiers \
  -X GET \
```

Successful Response

Response

```
[
  {
    "id": 1,
    "name": "Tier Name",
    "description": "Tier Description",
    "tier_id": "Tier Identifier",
    "enabled": true
  }
]
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| id | number | tier id - 1 |
| name | string | name of the tier |
| description | string | description of the tier |
| tier_id | string | identification of the tier |
| enabled | boolean | state of the tier |

## Get Tier

This call is used to return information about selected Tier for your merchant.

`GET /v1/fx/tiers/:id`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/fx/tiers/1 \
  -X GET \
```

Successful Response

Response

```
{
  "id": 1,
  "name": "Tier Name",
  "description": "Tier Description",
  "tier_id": "Tier Identifier",
  "enabled": true
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| id | number | tier id - 1 |
| name | string | name of the tier |
| description | string | description of the tier |
| tier_id | string | identification of the tier |
| enabled | boolean | state of the tier |

## Get Rates

This call is used to return all rates for Tier.

`GET /v1/fx/tiers/:tier_id/rates`

Note: `:tier_id` is the ID of Tier, not to be mistaken with `:tier_id` of the same entity.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/fx/tiers/1/rates \
  -X GET \
```

Successful Response

Response

```
[
  {
    "id": 2,
    "source_currency": "AUD",
    "target_currency": "EUR",
    "trading_rate": 0.0763
  }
]
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| id | number | rate id - 2 |
| source_currency | string | source currency of the rate |
| target_currency | string | target currency of the rate |
| trading_rate | float | trading rate |

---

# Get Rate

This call is used to return information about selected Rate for merchant. `GET /v1/fx/tiers/:tier_id/rates/:id`

Note: `:tier_id` is the ID of Tier, not to be mistaken with `:tier_id` of the same entity.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/fx/tiers/1/rates/1 \
  -X GET \
```

Successful Response

Response

```
{
  "id": 1,
  "source_currency": "CHF",
  "target_currency": "USD",
  "trading_rate": 1.0763
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| id | number | rate id - 1 |
| source_currency | string | source currency of the rate |
| target_currency | string | target currency of the rate |
| trading_rate | float | trading rate |

---

# Search Rate

This call is used to return information about selected Rate by currency pair.

`POST /v1/fx/tiers/:tier_id/rates/search`

Note: `:tier_id` is the ID of Tier, not to be mistaken with `:tier_id` of the same entity.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/fx/tiers/:tier_id/rates/search \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
  {
    "source_currency": "EUR",
    "target_currency": "USD"
  }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| source_currency | required | string | source currency |
| target_currency | required | string | target currency |

`required* = conditionally required`

Successful Response

Response

```
{
  "id": 1,
  "source_currency": "Tier Name",
  "target_currency": "Tier Description",
  "trading_rate": "Tier Identifier",
  "enabled": null
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| id | number | rate id - 1 |
| source_currency | string | source currency of the rate |
| target_currency | string | target currency of the rate |
| trading_rate | string | trading rate |

# Consumers

## Introduction

The Consumer entity brings Tokenization, Transactions and Web Payment Forms (WPF) together. It is a representation of a customer that can serve different purposes. A consumer is identified by providing both consumer ID and email. It is *explicitly* created via our Consumer API or *implicitly* by providing `customer_email` in either Transactions or WPF APIs. The main purpose of consumers is to group web payment forms and payment transactions. Using the merchant console, one can track consumers and find high-volume ones. The other role of consumers is to provide simplified, one-step tokenization of cardholder details. For Processing API that means securely storing card data in exchange for a token, which can be used for future payments. For WPF, customers can choose either to use previously stored cards or remember a new payment method.

## Consumer API

**CREATE CONSUMER**

Creates a consumer based on email address. Optionally, one can provide billing and shipping address. Addresses will be used, if none given, in Processing or WPF APIs.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/create_consumer/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <create_consumer_request>
   <email>consumer@email.com</email>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <shipping_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10001</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </shipping_address>
 </create_consumer_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| email | required | email address | The consumer email address must be unique. If another consumer exists with this email address, the request will be rejected. |
| **billing_address** | optional | | See Required vs Optional API params for details |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <create_consumer_response>
   <consumer_id>123456</consumer_id>
   <email>consumer@email.com</email>
   <status>enabled</status>
 </create_consumer_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| consumer_id | string(10) | Consumer unique reference |
| email | email address | Consumer email address |

| | | |
|---|---|---|
| status | string | Status of the consumer |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<create_consumer_response>
  <status>error</status>
  <code>330</code>
  <technical_message>Invalid email format!</technical_message>
  <message>Something went wrong, please contact support!</message>
</create_consumer_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the consumer |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

#### RETRIEVE CONSUMER

Retrieves consumer details based on consumer id or email.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/retrieve_consumer/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<retrieve_consumer_request>
  <consumer_id>123456</consumer_id>
</retrieve_consumer_request>'
```

Request

```
curl https://staging.gate.e-comprocessing.net/v1/retrieve_consumer/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<retrieve_consumer_request>
  <email>consumer@email.com</email>
</retrieve_consumer_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference. Required if the email is not provided |
| email | required | email address | Consumer email address. Required if the consumer_id is not provided |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<retrieve_consumer_response>
  <consumer_id>123456</consumer_id>
  <email>consumer@email.com</email>
  <status>enabled</status>
</retrieve_consumer_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| consumer_id | string(10) | Consumer unique reference |
| email | email address | Consumer email address |
| status | string | Status of the consumer |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<retrieve_consumer_response>
  <status>error</status>
  <code>702</code>
  <technical_message>Consumer not found!</technical_message>
  <message>Something went wrong, please contact support!</message>
</retrieve_consumer_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the consumer |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

#### UPDATE CONSUMER

Updates consumer email and addresses.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/update_consumer/ \
```

```
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<update_consumer_request>
  <consumer_id>123456</consumer_id>
  <email>consumer@email.com</email>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <shipping_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10001</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </shipping_address>
</update_consumer_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | email address | New email address |
| **billing_address** | optional | | See Required vs Optional API params for details |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |
| **shipping_address** | optional | | |
| first_name | optional | string(255) | Customer first name |
| last_name | optional | string(255) | Customer last name |
| address1 | optional | string(255) | Primary address |
| address2 | optional | string(255) | Secondary address |
| zip_code | optional | string | ZIP code |
| city | optional | string(255) | City |
| state | optional | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | optional | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<update_consumer_response>
  <consumer_id>123456</consumer_id>
  <email>consumer@email.com</email>
  <status>enabled</status>
</update_consumer_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| consumer_id | string(10) | Consumer unique reference |
| email | email address | Consumer email address |
| status | string | Status of the consumer |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<update_consumer_response>
  <status>error</status>
  <code>702</code>
  <technical_message>Consumer not found!</technical_message>
  <message>Something went wrong, please contact support!</message>
</update_consumer_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the consumer |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

| | | |
|---|---|---|
| message | string(255) | Human readable error message which can be displayed to users. |

#### DISABLE CONSUMER

Disable consumer from usage until further action.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/disable_consumer/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <disable_consumer_request>
   <consumer_id>123456</consumer_id>
   <email>consumer@email.com</email>
 </disable_consumer_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | email address | Consumer email address |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <disable_consumer_response>
   <consumer_id>123456</consumer_id>
   <email>consumer@email.com</email>
   <status>disabled</status>
 </disable_consumer_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| consumer_id | string(10) | Consumer unique reference |
| email | email address | Consumer email address |
| status | string | Status of the consumer |

Error Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <disable_consumer_response>
   <status>error</status>
   <code>702</code>
   <technical_message>Consumer not found!</technical_message>
   <message>Something went wrong, please contact support!</message>
 </disable_consumer_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the consumer |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

#### ENABLE CONSUMER

Enable consumer that was disabled in the past.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/enable_consumer/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <enable_consumer_request>
   <consumer_id>123456</consumer_id>
   <email>consumer@email.com</email>
 </enable_consumer_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | email address | Consumer email address |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <enable_consumer_response>
   <consumer_id>123456</consumer_id>
   <email>consumer@email.com</email>
   <status>enabled</status>
 </enable_consumer_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| consumer_id | string(10) | Consumer unique reference |

| | | |
|---|---|---|
| email | email address | Consumer email address |
| status | string | Status of the consumer |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<enable_consumer_response>
  <status>error</status>
  <code>702</code>
  <technical_message>Consumer not found!</technical_message>
  <message>Something went wrong, please contact support!</message>
</enable_consumer_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the consumer |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

#### GET CONSUMER CARDS

Get previously tokenized card details for a consumer.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/get_consumer_cards/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<get_consumer_cards_request>
  <consumer_id>123456</consumer_id>
  <email>consumer@email.com</email>
</get_consumer_cards_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | email address | Consumer email address |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<get_consumer_cards_response>
  <total>1</total>
  <card>
    <card_number>409603...0106</card_number>
    <card_holder>Travis Pastrana</card_holder>
    <expiration_month>12</expiration_month>
    <expiration_year>2026</expiration_year>
    <card_brand>master</card_brand>
  </card>
</get_consumer_cards_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| total | string(255) | Number of non-expired consumer cards |
| **card** | | |
| card_number | string(13..19) | Masked credit card number |
| card_holder | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| expiration_month | MM | Expiration month as printed on credit card |
| expiration_year | YYYY | Expiration year as printed on credit card |
| card_brand | string | Credit card brand |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<get_consumer_cards_response>
  <status>error</status>
  <code>702</code>
  <technical_message>Consumer not found!</technical_message>
  <message>Something went wrong, please contact support!</message>
</get_consumer_cards_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the consumer |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

# Managed Recurring

## Introduction

Managed Recurring provides the option to automatically schedule recurring transactions for a specific day and time. Managed Recurring is available after additional configuration.

## How to use managed recurring in Processing API

**REQUESTS**

Merchants can send managed recurring params in the request when creating **Init Recurring Sale** or **Init Recurring Sale3D** transactions via our Processing API.

Managed Recurring

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
   <card_holder>Travis Pastrana</card_holder>
   <cvv>834</cvv>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <managed_recurring>
     <mode>automatic</mode>
     <interval>days</interval>
     <first_date>2021-12-18</first_date>
     <time_of_day>5</time_of_day>
     <period>22</period>
     <amount>500</amount>
     <max_count>10</max_count>
   </managed_recurring>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **managed_recurring** | required | | |
| mode | required | String | Fill in with '**automatic**'. This indicates that the gateway will automatically manage the subsequent recurring transactions. |
| interval | optional | String(6) | The interval type for the period: **days** or **months**. The default value is **days** |
| first_date | optional | String(10) | Specifies the date of the first recurring event in the future, default value is date of creation + period. The format is ISO 8601 date format YYYY-MM-DD. |
| time_of_day | optional | Integer | Specifies the UTC hour in the day for the execution of the recurring transaction, default value 0. |
| period | required | Integer | Rebill period in days(30) or months(12). |
| amount | optional | Integer | Amount for the recurring transactions. |
| max_count | optional | Integer | Maximum number of times a recurring will occur. Omit this parameter for unlimited recurring. |

`required* = conditionally required`

## How to use managed recurring in WPF API

**REQUESTS**

Merchants can send managed recurring params in the request when creating **Init Recurring Sale** and **Init Recurring Sale3D** transactions via our WPF API.

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <wpf_payment>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <description>You are about to buy 3 shoes at www.shoes.com!</description>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <lifetime>60</lifetime>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
   <transaction_types>
     <transaction_type>
       <name>init_recurring_sale</name>
       <managed_recurring>
         <mode>automatic</mode>
         <interval>days</interval>
```

```
        <first_date>2021-12-18</first_date>
        <time_of_day>5</time_of_day>
        <period>22</period>
        <amount>500</amount>
        <max_count>10</max_count>
      </managed_recurring>
    </transaction_type>
  </transaction_types>
  <remember_card>true</remember_card>
</wpf_payment>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **managed_recurring** | required | | |
| mode | required | String | Fill in with '**automatic**'. This indicates that the gateway will automatically manage the subsequent recurring transactions. |
| interval | optional | String(6) | The interval type for the period: **days** or **months**. The default value is **days** |
| first_date | optional | String(10) | Specifies the date of the first recurring event in the future, default value is date of creation + period. The format is ISO 8601 date format YYYY-MM-DD. |
| time_of_day | optional | Integer | Specifies the UTC hour in the day for the execution of the recurring transaction, default value 0. |
| period | required | Integer | Rebill period in days(30) or months(12). |
| amount | optional | Integer | Amount for the recurring transactions. |
| max_count | optional | Integer | Maximum number of times a recurring will occur. Omit this parameter for unlimited recurring. |

`required* = conditionally required`

# Tokenization

## Introduction

Tokenization is the process of replacing sensitive cardholder data with a surrogate value ("token"). The data to be tokenized must include at least the primary account number (PAN).

Tokenization greatly reduces the sensitive data that businesses need to store, thus improving security of credit card transactions and minimizing the costs related to PCI DSS compliance.

We issue reversible non-cryptographic tokens to merchants via our Tokenization API and take care to store safely the tokenized cardholder data. Merchants are able to use the issued tokens instead of the cardholder data when creating credit card transactions via our Processing API. PCI DSS compliant merchants have also the possibility to exchange the token for the original cardholder data via our Tokenization API ("detokenization").

## Tokenization API

**ACCEPTED CARDHOLDER PARAMETERS**

All cardholder data parameters are accepted for tokenization - card number, cardholder, expiration year, expiration month. Please note - CVV is not accepted.

**CONSUMER REQUIRED**

An enabled consumer is required in order to use this API. You have to create one or use existing, please check Consumer API.

**TOKENIZE**

Tokenizes cardholder data and issues a corresponding token. Merchants should take care to save the plain-text token value in their system as once issued it is not possible to obtain it again. Attempting to tokenize the same cardholder data will issue a new token. The token can be used in the Processing API instead of the cardholder data.

> Please note, CVV can't be part of the cardholder data.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/tokenize/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <tokenize_request>
   <consumer_id>123456</consumer_id>
   <email>consumer@email.com</email>
   <token_type>uuid</token_type>
   <card_data>
     <card_number>4200000000000000</card_number>
     <card_holder>John Doe</card_holder>
     <expiration_month>05</expiration_month>
     <expiration_year>2026</expiration_year>
   </card_data>
 </tokenize_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | e-mail address | Consumer e-mail address |
| token_type | required | uuid | Token type format |
| **card_data** | required | | |
| card_number | required | string(13..19) | Complete credit/debit card number of customer |
| card_holder | optional | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| expiration_month | optional | MM | Expiration month as printed on credit card |
| expiration_year | optional | YYYY | Expiration year as printed on credit card |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <tokenize_response>
   <status>active</status>
```

```
  <token_id>34567</token_id>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <token_type>uuid</token_type>
  <consumer_id>123456</consumer_id>
</tokenize_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| token_id | string(32) | Unique token id |
| token | string(36) | Plain-text token value |
| token_type | uuid | Token type format |
| consumer_id | string(10) | Consumer unique reference |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<tokenize_response>
  <status>error</status>
  <code>720</code>
  <technical_message>Invalid token type!</technical_message>
  <message>Something went wrong, please contact support!</message>
</tokenize_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

#### DETOKENIZE

Exchanges the token with the tokenized cardholder data

Request

```
curl https://staging.gate.e-comprocessing.net/v1/detokenize/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<detokenize_request>
  <consumer_id>123456</consumer_id>
  <email>consumer@email.com</email>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <token_type>uuid</token_type>
</detokenize_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | e-mail address | Consumer e-mail address |
| token | required | string(36) | Plain-text token value |
| token_type | required | uuid | Token type format |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<detokenize_response>
  <status>active</status>
  <token_id>34567</token_id>
  <token_type>uuid</token_type>
  <card_data>
    <card_number>4200000000000000</card_number>
    <card_holder>John Doe</card_holder>
    <expiration_month>05</expiration_month>
    <expiration_year>2026</expiration_year>
  </card_data>
</detokenize_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| token_id | string(32) | Unique token id |
| token | string(36) | Plain-text token value |
| token_type | uuid | Token type format |
| **card_data** | | |
| card_number | string(13..19) | Complete credit/debit card number of customer |
| card_holder | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| expiration_month | MM | Expiration month as printed on credit card |
| expiration_year | YYYY | Expiration year as printed on credit card |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<detokenize_response>
  <status>error</status>
  <code>730</code>
  <technical_message>Invalid token!</technical_message>
  <message>Something went wrong, please contact support!</message>
</detokenize_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

#### UPDATE TOKEN

Updates the tokenized data corresponding to an existing valid token.

> Please note, PAN can't be updated

Request

```
curl https://staging.gate.e-comprocessing.net/v1/update_token/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<update_token_request>
  <consumer_id>123456</consumer_id>
  <email>consumer@email.com</email>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <token_type>uuid</token_type>
  <card_data>
    <card_number>4200000000000000</card_number>
    <card_holder>John Doe</card_holder>
    <expiration_month>05</expiration_month>
    <expiration_year>2026</expiration_year>
  </card_data>
</update_token_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | e-mail address | Consumer e-mail address |
| token | required | string(36) | Plain-text token value |
| token_type | required | uuid | Token type format |
| **card_data** | required | | |
| card_number | required | string(13..19) | Complete credit/debit card number of customer |
| card_holder | optional | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| expiration_month | optional | MM | Expiration month as printed on credit card |
| expiration_year | optional | YYYY | Expiration year as printed on credit card |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<update_token_response>
  <status>active</status>
  <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
  <token_type>uuid</token_type>
</update_token_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| token | string(36) | Plain-text token value |
| token_type | uuid | Token type format |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<update_token_response>
  <status>error</status>
  <code>730</code>
  <technical_message>Invalid token!</technical_message>
  <update_token_response>message'</update_token_response>
</update_token_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

| | | |
|---|---|---|
| message | string(255) | Human readable error message which can be displayed to users. |

#### VALIDATE TOKEN

Validates if a token is active, owned by a merchant, etc.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/validate_token/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <validate_token_request>
   <consumer_id>123456</consumer_id>
   <email>consumer@email.com</email>
   <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
   <token_type>uuid</token_type>
 </validate_token_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | e-mail address | Consumer e-mail address |
| token | required | string(36) | Plain-text token value |
| token_type | required | uuid | Token type format |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<validate_token_response>
  <status>active</status>
  <token_id>34567</token_id>
  <token_type>uuid</token_type>
</validate_token_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| token_id | string(32) | Unique token id |
| token_type | uuid | Token type format |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<validate_token_response>
  <status>error</status>
  <code>720</code>
  <technical_message>Invalid token type!</technical_message>
  <message>Something went wrong, please contact support!</message>
</validate_token_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

#### DELETE TOKEN

Deletes a token.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/delete_token/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <delete_token_request>
   <consumer_id>123456</consumer_id>
   <email>consumer@email.com</email>
   <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
   <token_type>uuid</token_type>
 </delete_token_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | e-mail address | Consumer e-mail address |
| token | required | string(36) | Plain-text token value |
| token_type | required | uuid | Token type format |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<delete_token_response>
  <status>active</status>
  <token_id>34567</token_id>
  <token_type>uuid</token_type>
</delete_token_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| token_id | string(32) | Unique token id |
| token_type | uuid | Token type format |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<delete_token_response>
  <status>error</status>
  <code>720</code>
  <technical_message>Invalid token type!</technical_message>
  <message>Something went wrong, please contact support!</message>
</delete_token_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

**GET CARD**

Exchanges the token with the tokenized masked cardholder data

Request

```
curl https://staging.gate.e-comprocessing.net/v1/get_card/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <get_card_request>
   <consumer_id>123456</consumer_id>
   <email>consumer@email.com</email>
   <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
   <token_type>uuid</token_type>
 </get_card_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| consumer_id | required | string(10) | Consumer unique reference |
| email | required | e-mail address | Consumer e-mail address |
| token | required | string(36) | Plain-text token value |
| token_type | required | uuid | Token type format |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<get_card_response>
  <status>active</status>
  <token_id>34567</token_id>
  <token_type>uuid</token_type>
  <card_data>
    <card_number>420000...0000</card_number>
    <card_holder>John Doe</card_holder>
    <expiration_month>05</expiration_month>
    <expiration_year>2026</expiration_year>
    <card_brand>visa</card_brand>
  </card_data>
</get_card_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| token_id | string(32) | Unique token id |
| token | string(36) | Plain-text token value |
| token_type | uuid | Token type format |
| **card_data** | | |
| card_number | string(13..19) | Complete credit/debit card number of customer |
| card_holder | string(255) | Full name of customer as printed on credit card (first name and last name at least) |
| expiration_month | MM | Expiration month as printed on credit card |
| expiration_year | YYYY | Expiration year as printed on credit card |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<detokenize_response>
  <status>error</status>
  <code>730</code>
  <technical_message>Invalid token!</technical_message>
  <message>Something went wrong, please contact support!</message>
</detokenize_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string | Status of the token |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

---

# How to tokenize cardholder data in Processing API

In order to tokenize card details you need to set the `remember_card` flag to "true".

#### CREATE A CONSUMER

Please provide `customer_email` in addition to cardholder details and the `remember_card` flag. This will create a consumer and tokenize cardholder details. `consumer_id` and `token` will be returned in the response and notification. `consumer_id` will be returned in the reconcile response.

Request

```
curl https://staging.gate.e-comprocessing.net/process/TERMINAL_TOKEN/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <remember_card>true</remember_card>
 </payment_transaction>'
```

#### USE EXISTING CONSUMER

Please provide `consumer_id` and `customer_email` in addition to cardholder details and the `remember_card` flag. An existing consumer will be used, if identified. `consumer_id` and `token` will be returned in the response and notification. `consumer_id` will be returned in the reconcile response.

Request

```
curl https://staging.gate.e-comprocessing.net/process/TERMINAL_TOKEN/ \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <card_holder>Travis Pastrana</card_holder>
   <card_number>4200000000000000</card_number>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <remember_card>true</remember_card>
   <consumer_id>123456</consumer_id>
 </payment_transaction>'
```

---

# How to use tokens in Processing API

#### REQUESTS

Merchants can substitute cardholder parameters with tokens in the request when creating credit card transactions via our Processing API. Please note: merchants may choose to tokenize all cardholder parameters or only a subset. In the latter case the remaining parameters would need to be provided in the request. An enabled consumer is required to use tokens in Processing API. Please provide `consumer_id` and `customer_email` along with the `token`.

All Cardholder Parameters Have Been Tokenized

Request

```
curl https://staging.gate.e-comprocessing.net/process/TERMINAL_TOKEN/ \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
   <cvv>834</cvv>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <consumer_id>123456</consumer_id>
 </payment_transaction>'
```

Only Pan Has Been Tokenized

Request

```
curl https://staging.gate.e-comprocessing.net/process/TERMINAL_TOKEN/ \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>sale</transaction_type>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
   <card_holder>Travis Pastrana</card_holder>
   <cvv>834</cvv>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <consumer_id>123456</consumer_id>
 </payment_transaction>'
```

---

# How to tokenize cardholder data in WPF API

In order to offer saving a payment method for future use, set the `remember_card` flag to "true".

#### CREATE A CONSUMER

Please provide `customer_email` in addition to the `remember_card` flag. This will create a consumer and offer the user to save cardholder details (tokenize). `consumer_id` will be returned in the response and reconcile response. `consumer_id` and `token` will be returned in the notification.

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <wpf_payment>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <description>You are about to buy 3 shoes at www.shoes.com!</description>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <lifetime>60</lifetime>
   <billing_address>
     <first_name>Travis</first_name>
     <last_name>Pastrana</last_name>
     <address1>Muster Str. 12</address1>
     <zip_code>10178</zip_code>
     <city>Los Angeles</city>
     <neighborhood>Hollywood</neighborhood>
     <state>CA</state>
     <country>US</country>
   </billing_address>
   <risk_params>
     <user_id>123456</user_id>
   </risk_params>
   <transaction_types>
     <transaction_type>authorize</transaction_type>
     <transaction_type>sale</transaction_type>
   </transaction_types>
   <remember_card>true</remember_card>
 </wpf_payment>'
```

#### USE EXISTING CONSUMER

Please provide `consumer_id` and `customer_email` in addition to the `remember_card` flag. An existing consumer will be used, if identified, and offer the user to save cardholder details (tokenize). `consumer_id` will be returned in the response and reconcile response. `consumer_id` and `token` will be returned in the notification.

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
   -X POST \
   -H "Content-Type: text/xml" \
   -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <wpf_payment>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <description>You are about to buy 3 shoes at www.shoes.com!</description>
   <notification_url>https://www.example.com/notification</notification_url>
   <return_success_url>http://www.example.com/success</return_success_url>
   <return_failure_url>http://www.example.com/failure</return_failure_url>
   <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
   <amount>100</amount>
   <currency>USD</currency>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <lifetime>60</lifetime>
   <billing_address>
```

```
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
  <transaction_types>
    <transaction_type>authorize</transaction_type>
    <transaction_type>sale</transaction_type>
  </transaction_types>
  <remember_card>true</remember_card>
  <consumer_id>123456</consumer_id>
</wpf_payment>'
```

## How to use tokens in WPF API

Please provide `consumer_id` and `customer_email`. An existing consumer will be used, if identified, and offer the user to select a previously stored card for payment.

Request

```
curl https://staging.wpf.e-comprocessing.net/wpf \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<wpf_payment>
  <transaction_id>119643250547501c79d8295</transaction_id>
  <usage>40208 concert tickets</usage>
  <description>You are about to buy 3 shoes at www.shoes.com!</description>
  <notification_url>https://www.example.com/notification</notification_url>
  <return_success_url>http://www.example.com/success</return_success_url>
  <return_failure_url>http://www.example.com/failure</return_failure_url>
  <return_cancel_url>http://www.example.com/cancel.html</return_cancel_url>
  <amount>100</amount>
  <currency>USD</currency>
  <customer_email>travis@example.com</customer_email>
  <customer_phone>+1987987987987</customer_phone>
  <lifetime>60</lifetime>
  <billing_address>
    <first_name>Travis</first_name>
    <last_name>Pastrana</last_name>
    <address1>Muster Str. 12</address1>
    <zip_code>10178</zip_code>
    <city>Los Angeles</city>
    <neighborhood>Hollywood</neighborhood>
    <state>CA</state>
    <country>US</country>
  </billing_address>
  <risk_params>
    <user_id>123456</user_id>
  </risk_params>
  <transaction_types>
    <transaction_type>authorize</transaction_type>
    <transaction_type>sale</transaction_type>
  </transaction_types>
  <consumer_id>123456</consumer_id>
</wpf_payment>'
```

## Supported transaction types

All transaction types accepting cardholder data are supported - Account Verification, Authorize, Authorize3D, Sale, Sale3D, InitRecurringSale, InitRecurringSale3D, Payout, Intersolve, Fashioncheque.

## Importation of external tokens and card details

**INPUT CSV FILE FORMAT**

**CSV file headers:**

| Header | req? | Description |
|---|---|---|
| token | required | Plain-text token value (merchant/PSP token to migrate to us) |
| token_type | optional | Token type format |
| status | optional | Status of the token |
| card_number | required | Credit card number |
| card_holder | required | Full name of customer as printed on cc (first name and last name at least) |
| expiration_month | required | Expiration month as printed on credit card |
| expiration_year | required | Expiration year as printed on credit card |
| email | required | Customer email |
| first_name | optional | Customer first name used for billing address details |
| last_name | optional | Customer last name used for billing address details |
| address1 | optional | Customer primary address used for billing address details |
| address2 | optional | Customer secondary address used for billing address details |
| zip_code | optional | Zip code used for billing address details |
| city | optional | Customer city used for billing address details |
| state | optional | State code in ISO 3166-2, used for billing address details |
| country | optional | Country code in ISO 3166 used for billing address details |

CSV file delimiter must be ',' (comma).

#### ENCRYPTION OF THE CSV FILE

The CSV contains sensitive data and therefore it must be encrypted. A public GPG key should be used for this purpose.

Command to encrypt the file. It will generate encrypted file ending with .gpg extension:

```
gpg --encrypt --recipient test@email.com test_token_file.csv
```

#### UPLOADING THE ENCRYPTED CSV FILE TO REMOTE SFTP SERVER

Once the CSV file is encrypted it must be uploaded to our remote SFTP server for processing and importing its content.

The remote SFTP connection has the following options which should be provided on demand (please, contact tech-support@e-comprocessing.com for more details):

**host** - host where the remote SFTP server is located (ex. prod.sftp.e-comprocessing.net)
**user** - user allowed to connect to the remote SFTP server (ex. IMPORT_CARDS_PROD_USER)
**port** - port listening for SFTP connections (ex. 22)
**directory** - directory to upload encrypted CSV files (ex. imported_cards/)
**ssh_key** - SSH key to authenticate the relevant user

Commands to upload the encrypted CSV file to remote SFTP server:

```
sftp -i "path-to-the-ssh-key-file" -P 22 IMPORT_CARDS_PROD_USER@prod.sftp.e-comprocessing.net:imported_cards/
```

Once connected to the remote SFTP:

```
sftp> put test_tokens_file.csv.gpg
Uploading test_tokens_file.csv.gpg to /imported_cards/test_tokens_file.csv.gpg
sftp> quit
```

#### DOWNLOADING A RESPONSE CSV FILE FROM THE REMOTE SFTP SERVER

Once the tokens are imported in Genesis together with the relevant card details, a response CSV file would be generated. It will contain mapping between the tokens and the consumer ids and emails associated with them. They would be needed for successful usage of the imported tokens.

#### RESPONSE CSV FILE FORMAT

**CSV file headers:**

| Header | Type | Description |
|---|---|---|
| token | string(36) | Plain-text newly generated token value |
| consumer_id | string(10) | ID of the created consumer entity |
| old_token | string | Plain-text old token value (merchant/PSP token passed to us for migration) |
| token_id | string(32) | ID for the newly generated token |
| token_type | uuid | Token type format |
| status | string | Status of the token |

Commands to retrieve the response CSV files from the remote SFTP server:

```
sftp -i "path-to-the-ssh-key-file" -P 22 IMPORT_CARDS_PROD_USER@prod.sftp.e-comprocessing.net:exchange/
```

Once connected to the remote SFTP:

```
sftp> get test_tokens_file_response.csv
Fetching /exchange/test_tokens_file_response.csv to test_tokens_file_response.csv
sftp> quit
```

# Transaction API

## Transaction Card Expiry Date Update API

Each card-based transaction has an expiration date, which can be updated using the Transaction Card Expiry Date Update API.

The API endpoint is suitable for recurring payments where the card has been renewed and has now a different expiration date.

Request

```
curl https://staging.gate.e-comprocessing.net/v1/transaction/expiry_date/:transaction_unique_id \
  -X PUT \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <update_card_expiration_request>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
 </update_card_expiration_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| expiration_month | required | MM | Expiration month as printed on credit card |
| expiration_year | required | YYYY | Expiration year as printed on credit card |

`required* = conditionally required`

The provided expiration date must be in the future and after the current expiration date of the payment transaction

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<update_card_expiration_response>
  <status>success</status>
  <code>200</code>
  <message>Card expiry date updated successfully!</message>
</update_card_expiration_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the response |
| code | integer | Successful code (200) |
| message | string(255) | Human readable error message which can be displayed to users. |

Error Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<update_card_expiration_response>
  <status>error</status>
  <code>300</code>
  <technical_message>The provided expiration date must be in the future and after the current one</technical_message>
  <message>Please check input data for errors!</message>
</update_card_expiration_response>
```

**Error Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the response |
| code | integer | Error code according to Error code table |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |
| message | string(255) | Human readable error message which can be displayed to users. |

# APM Services

## Alipay Register Merchant

## Introduction

Alipay Register Merchant call allows to register merchants of online payments into Alipay system or to update the registration information of merchants.

To interact with the Alipay Register Merchant API, you need to provide login credentials using standard HTTP Basic Authentication. Be sure to set Content-type: application/json in your headers. Replace terminal_id in the request with the desired terminal ID.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/alipay/register_merchant/:terminal_token \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "merchant_type": "ENTERPRISE",
   "site_information": {
     "type": "APP",
     "url": "https://example.com",
     "name": "Example"
   },
   "contact_number": "1234567",
   "registration_number": "1234567",
   "representative_id": "1234567",
   "representative_name": "Your Name"
 }'
```

Successful Response

Response

```
 {
   "success": "true"
 }
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/alipay/register_merchant/:terminal_token \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "merchant_type": "INDIVIDUAL",
   "site_information": {
     "type": "WEB",
     "url": "https://example.com",
     "name": "Example"
   },
   "contact_number": "1234567",
   "registration_number": "1234567",
   "shareholder_id": "1234567",
   "shareholder_name": "Your Name"
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| merchant_type | required | string | Merchant type, the value can be INDIVIDUAL for the sole proprietorship or ENTERPRISE. |
| site_information | required | object | This field is in JSON object. See site_infos for details. |
| shareholder_id | required* | string(128) | ID or passport number, or business registration number of the primary shareholder of the merchant. Specify this field only when the merchant type is ENTERPRISE. |
| shareholder_name | required* | string(64) | Legal name of the primary shareholder of the merchant. Specify this field only when the merchant type is ENTERPRISE. |
| registration_number | required* | string(128) | Business registration number specified on the business registration document. Note: This field is not required when the merchant type is INDIVIDUAL and no registration number exists. |
| representative_name | required* | string(64) | Full legal name of the business owner. Specify this field only when the merchant type is INDIVIDUAL. This field is optional if the merchant type is ENTERPRISE. |
| representative_id | required* | string(128) | ID or passport number of the business owner. Specify this field only when the merchant type is INDIVIDUAL. This field is optional if the merchant type is ENTERPRISE. |
| settlement_number | optional | string(64) | Settlement bank account number of the merchant. Use letters and numbers only. |
| contact_number | required | string(64) | Contact phone number of the merchant, numbers and special characters +-() only. |
| contact_email | optional | string(128) | Contact email address of the merchant. |
| customer_service_number | optional | string(64) | Customer service phone number of the merchant, numbers and special characters +-() only. |
| customer_service_email | optional | string(128) | Customer service email address of the merchant. |

`required* = conditionally required`

**ALIPAY REGISTER MERCHANT SITE INFO PARAMETERS**

---

## Site info

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| type | required | string | Site type. For website URL, the value of this parameter must be WEB. For app download URL, the value of this parameter must be APP. Use uppercase. |
| url | required | string(256) | Site URL. When site_type is WEB, pass the URL in this format: http/https + SLD + TLD. When site_type is APP, pass the APP download URL starting with http/https. |
| name | optional | string(512) | Site name. |

`required* = conditionally required`

---

## Klarna

---

## Introduction

Klarna Services provides the ability for merchants to release remaining authorization, resend invoice, update order items or to update order address of a transaction.

Klarna services requests are handled exactly like transaction requests via XML and authentication is required. Note that Klarna services can be done via transaction_id

**RELEASE REMAINING AUTHORIZATION API**

The URL for Release Remaining Authorization API is:

Production:

`https://gate.e-comprocessing.net/klarna/release_remaining_authorization`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/klarna/release_remaining_authorization`

Release Remaining Authorization service is used for approved Invoice transaction with at least one approved partial Invoice Capture transaction.
With this call the remaining authorization amount will be released and Invoice Captures will be forbidden for the Invoice transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/klarna/release_remaining_authorization \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <release_remaining_authorization_request>
   <transaction_id>a1qf12e81eb23d0e00ffb85b1db7d152</transaction_id>
 </release_remaining_authorization_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_id | required | string(255) | Unique transaction id defined by merchant |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <release_remaining_authorization_request>
   <status>success</status>
   <technical_message>Transaction operation successful!</technical_message>
 </release_remaining_authorization_request>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

#### RESEND INVOICE API

The URL for Resend Invoice API is:

Production:

`https://gate.e-comprocessing.net/klarna/resend_invoice`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/klarna/resend_invoice`

> Resend Invoice service is used only for approved not yet refunded Invoice Capture transaction.
> With this call Klarna will resend the invoice of the captured transaction.

Request

```
curl https://staging.gate.e-comprocessing.net/klarna/resend_invoice \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <resend_invoice_request>
   <transaction_id>a1qf12e81eb23d0e00ffb85b1db7d152</transaction_id>
 </resend_invoice_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_id | required | string(255) | Unique transaction id defined by merchant |

`required* = conditionally required`

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <resend_invoice_request>
   <status>success</status>
   <technical_message>Transaction operation successful!</technical_message>
 </resend_invoice_request>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

#### UPDATE ORDER ITEMS API

The URL for Update Order Items API is:

Production:

`https://gate.e-comprocessing.net/klarna/update_order_items`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/klarna/update_order_items`

> Update Order Items service is used only for approved but not yet captured Invoice transaction.
> With this call amount and associated items will be updated.

Request

```
curl https://staging.gate.e-comprocessing.net/klarna/update_order_items \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <update_order_items_request>
   <transaction_id>a1qf12e81eb23d0e00ffb85b1db7d152</transaction_id>
   <amount>10000</amount>
   <items>
     <item>
       <type>physical</type>
       <reference>19-402-BG1</reference>
       <name>BatteryPowerPack</name>
       <quantity>1</quantity>
       <unit_price>5000</unit_price>
       <tax_rate>0</tax_rate>
       <total_amount>5000</total_amount>
       <total_discount_amount>0</total_discount_amount>
       <total_tax_amount>0</total_tax_amount>
     </item>
   </items>
 </update_order_items_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| **items** | required | | List with items |
| item_type | required | string(255) | Order line type. Possible values: Supported item types |
| quantity | required | integer | Non-negative. The item quantity |
| unit_price | required | integer | Minor units. Includes tax, excludes discount(max value: 100000000) |
| total_amount | required | integer | |

| | | | |
|---|---|---|---|
| | | | Includes tax and discount. Must match (quantity unit price) - total discount amount divided by quantity (max value: 100000000) |
| reference | optional | string(255) | Article number, SKU or similar |
| name | optional | string(255) | Descriptive item name |
| tax_rate | optional | integer | Non-negative. In percent, two implicit decimals. I.e 2500 = 25.00 percent |
| total_discount_amount | optional | integer | Non-negative minor units. Includes tax |
| total_tax_amount | optional | integer | Must be within 1 of total amount - total_amount * 10000 / (10000 + tax rate). Negative when type is discount |
| image_url | optional | url | URL to an image that can be later embedded in communications between Klarna and the customer. (max 1024 characters) |
| product_url | optional | url | URL to an image that can be later embedded in communications between Klarna and the customer. (max 1024 characters) |
| quantity_unit | optional | string(8) | Unit used to describe the quantity, e.g. kg, pcs... If defined has to be 1-8 characters |
| **product_identifiers** | optional | | List with product identifiers |
| brand | optional | string(255) | The product's brand name as generally recognized by consumers. If no brand is available for a product, do not supply any value |
| category_path | optional | string(255) | The product's category path as used in the merchant's webshop. Include the full and most detailed category and separate the segments with ' > ' |
| global_trade_item_number | optional | string(255) | The product's Global Trade Item Number (GTIN). Common types of GTIN are EAN, ISBN or UPC. Exclude dashes and spaces, where possible |
| manufacturer_part_number | optional | string(255) | The product's Manufacturer Part Number (MPN), which - together with the brand - uniquely identifies a product. Only submit MPNs assigned by a manufacturer and use the most specific MPN possible |
| **merchant_data** | optional | | List with merchant data |
| marketplace_seller_info | optional | string(255) | Information for merchant marketplace |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<update_order_items_request>
  <status>success</status>
  <technical_message>Transaction operation successful!</technical_message>
</update_order_items_request>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

#### UPDATE ORDER ADDRESS API

The URL for Update Order Address API is:

Production:

`https://gate.e-comprocessing.net/klarna/update_order_address`

Staging (for integration):

`https://staging.gate.e-comprocessing.net/klarna/update_order_address`

Update Order Address service is used only for approved but not yet captured Klarna transaction.
With this call billing and shipping addresses will be updated.

Request

```
curl https://staging.gate.e-comprocessing.net/klarna/update_order_address \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <update_order_address_request>
   <transaction_id>a1qf12e81eb23d0e00ffb85b1db7d152</transaction_id>
   <billing_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Berlin</city>
     <neighborhood>Lichtenberg</neighborhood>
     <country>DE</country>
   </billing_address>
   <shipping_address>
     <first_name>Barney</first_name>
     <last_name>Rubble</last_name>
     <address1>14, Nerazdelni str</address1>
     <zip_code>1407</zip_code>
     <city>Berlin</city>
     <neighborhood>Lichtenberg</neighborhood>
     <country>DE</country>
   </shipping_address>
 </update_order_address_request>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| transaction_id | required | string(255) | Unique transaction id defined by merchant |
| **billing_address** | required | | See Required vs Optional API params for details |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |

| | | | |
|---|---|---|---|
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |
| **shipping_address** | required | | |
| first_name | required* | string(255) | Customer first name |
| last_name | required* | string(255) | Customer last name |
| address1 | required* | string(255) | Primary address |
| address2 | required* | string(255) | Secondary address |
| zip_code | required* | string | ZIP code |
| city | required* | string(255) | City |
| neighborhood | required* | string(255) | Neighborhood |
| state | required* | string(2) | State code in ISO 3166-2, required for USA and Canada |
| country | required | string(2) | Country code in ISO 3166 |

`required* = conditionally required`

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<update_order_address_request>
  <status>success</status>
  <technical_message>Transaction operation successful!</technical_message>
</update_order_address_request>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| status | string(255) | Status of the transaction, see states |
| technical_message | string(255) | Technical error message (for internal use only, not to be displayed to users). |

---

# Trustly Select Account

---

# Introduction

Trustly Select Account call allows the consumer to select their Trustly account, which is later on used to complete a Trustly Bank Pay-out transaction.

A typical flow is:

The merchant makes an API call to SelectAccount and redirects the consumer to the url.

The consumer selects his/her bank and completes the identification process.

The consumer is redirected back to the merchant at the SuccessURL, note that the account might not be verified yet at this point.

When the account is verified, Trustly sends an account notification ( Account_ID) to the merchant's system with information about the selected account and Account_id unique for each user's bank account.

> To interact with the Trustly Select Account API, you need to provide login credentials using standard HTTP Basic Authentication. Be sure to set Content-type: application/json in your headers. Replace terminal_id in the request with the desired terminal ID.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/trustly/select_account/:terminal_token \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "country": "UK",
   "first_name": "Travis",
   "last_name": "Pastrana",
   "ip_address": "255.255.255.255",
   "mobile_phone": "+441509813888",
   "national_id": "8910103344",
   "birth_date": "1989-10-10",
   "success_url": "https://example.com/success",
   "failure_url": "https://example.com/failure",
   "user_id": "12345678",
   "unique_id": "gp634ec5e7dbe6ca3871974accb875cd",
   "locale": "en_US"
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| country | required | string(2) | Country code in ISO 3166 |
| first_name | required | string(255) | Customer first name |
| ip_address | required | string(255) | IP address of the customer |
| email | required | e-mail address | Must contain valid e-mail of the customer |
| mobile_phone | required | string(32) | Must contain valid phone number of the customer |
| national_id | required | string(20) | National Identifier number of the customer |
| birth_date | required | yyyy-mm-dd | Must contain valid birth date of the customer |
| success_url | required | url | URL where the customer is sent to after successful authentication |
| failure_url | required | url | URL where the customer is sent to after failed authentication |

| | | | |
|---|---|---|---|
| user_id | required | string(255) | Unique user identifier defined by merchant in their own system. ID, username, hash or anything uniquely identifying the consumer requesting the deposit. Must be static per each consumer for any type of transaction where this consumer is involved (trustly_sale, bank pay_out, register_account, select account). |
| unique_id | required | string(255) | Unique identifier defined by merchant |
| locale | required | string(20) | Customer's localization preference in the format [language[_territory]]. Language is the ISO 639 code and the territory ISO 3166 code. |

`required* = conditionally required`

Successful Response

Response

```
{
  "order_id": "4014321",
  "url": "http://example.com/customer_redirect"
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| order_id | string(255) | Order identifier |
| url | url | URL where user has to be redirected to complete select account process. |

---

# Trustly Register Account

---

# Introduction

Trustly Register Account call allows the merchant to verify customer's bank account details and get the associated account id which can be saved and used for future payouts.

To interact with the Trustly Register Account API, you need to provide login credentials using standard HTTP Basic Authentication. Be sure to set Content-type: application/json in your headers. Replace terminal_id in the request with the desired terminal ID.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/trustly/register_account/:terminal_token \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "first_name": "Travis",
   "last_name": "Pastrana",
   "mobile_phone": "+441509813888",
   "national_id": "8910103344",
   "birth_date": "1989-10-10",
   "user_id": "12345678",
   "clearing_house": "SPAIN",
   "account_number": "ES8701820004756386447000",
   "bank_number": ""
 }'
```

**List of supported clearinghouses:**

AUSTRIA, BELGIUM, BULGARIA, CROATIA, CYPRUS, CZECH_REPUBLIC, DENMARK, ESTONIA, FINLAND, FRANCE, GERMANY, GREECE, HUNGARY, IRELAND, ITALY, LATVIA, LITHUANIA, LUXEMBOURG, MALTA, NETHERLANDS, NORWAY, POLAND, PORTUGAL, ROMANIA, SLOVAKIA, SLOVENIA, SPAIN, SWEDEN, UNITED_KINGDOM

Successful Response

Response

```
{
  "account_id": "1044806532",
  "clearing_house": "SPAIN",
  "bank": "BBVA",
  "descriptor": "******************447000"
}
```

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/trustly/register_account/:terminal_token \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "first_name": "Travis",
   "last_name": "Pastrana",
   "mobile_phone": "+441509813888",
   "national_id": "123456789",
   "birth_date": "1989-10-10",
   "user_id": "12345678",
   "clearing_house": "SWEDEN",
   "account_number": "8257466",
   "bank_number": "5839"
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| first_name | required | string(255) | Customer first name |
| last_name | required | string(255) | Customer last name |
| email | optional | e-mail address | Must contain valid e-mail of the customer |
| mobile_phone | optional | string(32) | Must contain valid phone number of the customer |
| national_id | optional | string(20) | National Identifier number of the customer |
| birth_date | required | yyyy-mm-dd | Must contain valid birth date of the customer |
| user_id | required | string(255) | Unique user identifier defined by merchant in their own system. ID, username, hash or anything uniquely identifying the consumer requesting the deposit. Must be static per each consumer for any type of transaction where this consumer is involved (trustly_sale, bank pay_out, register_account, select account). |

| | | | |
|---|---|---|---|
| clearing_house | required | string(255) | The clearing house of the customer's bank account. Typically the name of a country in uppercase letters. |
| account_number | required | string(32) | The account number of the customer's bank account. Can be either IBAN or country-specific format. |
| bank_number | required | string(32) | The bank number of the customer's account in the given clearing house. For bank accounts in IBAN format, just provide an empty string ("") |

`required* = conditionally required`

List of clearing houses with IBAN support: AUSTRIA, BELGIUM, BULGARIA, CROATIA, CYPRUS, CZECH_REPUBLIC, DENMARK, ESTONIA, FINLAND, FRANCE, GERMANY, GREECE, HUNGARY, IRELAND, ITALY, LATVIA, LITHUANIA, LUXEMBOURG, MALTA, NETHERLANDS, NORWAY, POLAND, ROMANIA, SLOVAKIA, SLOVENIA, SPAIN, UNITED_KINGDOM

List of clearing houses with account number and bank number support: CZECH_REPUBLIC, DENMARK, ESTONIA, HUNGARY, NORWAY, PORTUGAL, SWEDEN, UNITED_KINGDOM

Successful Response

Response

```
{
  "account_id": "7653385737",
  "clearing_house": "SWEDEN",
  "bank": "Skandiabanken",
  "descriptor": "*******4057"
}
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| account_id | string(255) | Unique identifier associated with the account |
| clearing_house | string(255) | The clearing house of the account |
| bank | string(255) | Name of the bank for this account |
| descriptor | string(255) | A descriptor for this account |

The account_id received in the response should be stored to process further payout requests.

# TransferTo

# Introduction

TransferTo API endpoint provides merchants with the ability to retrieve an up-to-date list of TransferTo Payers. Those are the institutions that provide the money to the consumers.

TransferTo Retrieve Payers request needs authentication.

#### RETRIEVE PAYERS API

This request is used to retrieve up-to-date TransferTo Payers list.

`GET /transfer_to_payers/payers`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/transfer_to_payers/payers \
  -X GET \
```

Successful Response

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<payers_response>
  <payers>
    <payer>
      <id>1</id>
      <name>Sample Payer 1</name>
      <currency>USD</currency>
      <country_iso_code>USA</country_iso_code>
      <service>BankAccount</service>
      <transfer_type>B2C</transfer_type>
      <transaction_requirements>{"sample":"requirements"}</transaction_requirements>
    </payer>
    <payer>
      <id>2</id>
      <name>Sample Payer 2</name>
      <currency>EUR</currency>
      <country_iso_code>FRA</country_iso_code>
      <service>MobileWallet</service>
      <transfer_type>B2C</transfer_type>
      <transaction_requirements>{"sample":"requirements"}</transaction_requirements>
    </payer>
  </payers>
</payers_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| **payer** | | |
| id | string | ID of the Payer (used as payer_id in the TransferToPayout transaction) |
| name | string | Name of the Payer |
| currency | string(3) | The currency in which the Payer provides the money |
| country_iso_code | string(3) | The ISO code of the country in which the Payer operates |
| service | string | The service used by the Payer to provide the money |
| transfer_type | string(3) | The money transfer type. Can be `B2B`, `B2C` or `C2C`. |

| | | |
|---|---|---|
| transaction_requirements | string(1024) | JSON containing the Payer's requirements |

# Errors

## Error groups table

| Code | Name | Description |
|---|---|---|
| (100..199) | Systems errors | Transaction could not be processed and was not passed to issuer. |
| (200..299) | Communication errors | Transaction could not be processed properly. Issuer could not be reached or returned invalid data. Errors 230 - 250 need to be reconciled as they might have been processed properly issuer-wise. |
| (300..399) | Input data errors | Transactions cannot be processed due to invalid incoming data in your request. |
| (400..499) | Workflow errors | Workflow errors will occur if you trigger a transaction that is not possible at this time in the workflow, e.g. a refund on a declined transaction. |
| (500..599) | Processing errors | These errors occur when a transaction was declined by the issuer. |
| (600..699) | Risk errors | Risk errors occur when any of the risk management systems will not let the transaction pass through. |
| (700..799) | Tokenization errors | Tokenization related errors. |
| (800..899) | Genesis KYC Services errors | Genesis KYC Services errors |
| (900..999) | Issuer errors | Issuer errors occur when the issuer is unreachable or has other technical problems. If you experience this kind of errors, contact support. |

## Error codes tables

**System Errors**

| Code | Name | Description |
|---|---|---|
| 100 | SystemError | A general system error occurred |
| 101 | MaintenanceError | System is undergoing maintenance, request could not be handled. |
| 110 | AuthenticationError | Login failed. Check your API credentials. |
| 120 | ConfigurationError | Configuration error occurred, e.g. terminal not config- ured properly. Check terminal settings. |

**Communication Errors**

| Code | Name | Description |
|---|---|---|
| 200 | CommunicationError | Communication with issuer failed, please contact sup- port. |
| 210 | ConnectionError | Connection to issuer could not be established, please contact support. |
| 220 | AccountError | Issuer account data invalid, please contact support. |
| 230 | TimeoutError | Issuer does not respond within given timeframe - please reconcile |
| 240 | ResponseError | Issuer returned invalid response - please reconcile and contact support |
| 250 | ParsingError | Issuer response could not be parsed - please reconcile and contact support. |

**Input Data Errors**

| Code | Name | Description |
|---|---|---|
| 300 | InputDataError | Invalid were data sent to the API. |
| 310 | InvalidTransactionTypeError | Invalid transaction type was passed to API. See transaction types. |
| 320 | InputDataMissingError | Required argument is missing. Check parameters. |
| 330 | InputDataFormatError | Argument passed in invalid format. Check parame- ters. |
| 340 | InputDataInvalidError | Argument passed in valid format but makes no sense (e.g. incorrect country code or currency). Check parameters. |
| 350 | InvalidXmlError | The input XML could not be parsed due to invalid code. Please check XML data. |
| 360 | InvalidContentTypeError | Missing or invalid content type: should be text/xml! |

**Workflow Errors**

| Code | Name | Description |
|---|---|---|
| 400 | WorkflowError | A transaction was triggered that is not possible at this time in the workflow, e.g. a refund on a declined transaction. |
| 410 | ReferenceNotFoundError | Reference transaction was not found. |
| 420 | ReferenceWorkflowError | Wrong Workflow specified. |
| 430 | ReferenceInvalidatedError | Reference transaction already invalidated! |
| 440 | ReferenceMismatchError | Data mismatch with reference, e.g. amount exceeds reference |
| 450 | DoubletTransactionError | Transaction doublet was detected, transaction was blocked. This happens if several transactions with same amount, cardholder, cc number, cvv and expiry date are sent within 5 minutes. |
| 460 | TransactionNotFoundError | The referenced transaction could not be found. |
| 470 | ChargebackNotFoundError | Chargeback not found! |
| 471 | RapidDisputeResolutionNotFoundError | Rapid Dispute Resolution not found! |
| 480 | RetrievalRequestNotFoundError | Retrieval Request not found! |
| 490 | FraudReportNotFoundError | Fraud Report not found! |

**Processing Errors**

| Code | Name | Description |
|---|---|---|
| 500 | ProcessingError | Transaction declined by issuer |
| 510 | InvalidCardError | Transaction declined, Credit card number is invalid. |
| 511 | IssuerOctNotEnabledError | OCT not enabled error. |
| 520 | ExpiredCardError | Transaction declined, expiration date not in the future or date invalid. |
| 530 | TransactionPendingError | Transaction pending. |
| 540 | CreditExceededError | Amount excceds credit card limit. |
| 550 | IssuingError | The voucher could not be issued! |
| 551 | ScaRequiredError | SCA required! |

**Risk Errors**

| Code | Name | Description |
|---|---|---|
| 600 | RiskError | Transaction declined by risk management |
| 601 | InterchangeRejectError | Interchange reject received for transaction! |
| 609 | BinCountryCheckError | Card bin does not match billing country |
| 610 | CardBlacklistError | Card is blacklisted |
| 611 | BinBlacklistError | BIN blacklisted. |
| 612 | CountryBlacklistError | Country blacklisted. |
| 613 | IpBlacklistError | IP address blacklisted. |
| 614 | BlacklistError | value from payment transaction or risk params is blacklisted. |
| 615 | CardWhitelistError | PAN Whitelist Filter blocked the transaction. This filter - like the above one - uses the PAN blacklist (BL) to perform CC number checks against the BL in the DB. This filter however will reject transactions from a CC with a number which is not whitelisted. |
| 620 | CardLimitExceededError | Card limit exceeded configured limits. |
| 621 | TerminalLimitExceededError | Terminal limits exceeded. |
| 622 | ContractLimitExceededError | MID limits exceeded. |
| 623 | CardVelocityExceededError | Velocity by unknown card exceeded! |
| 624 | CardTicketSizeExceededError | Ticketsize by unknown card exceeded! |
| 625 | UserLimitExceededError | User limit exceeded configured limits. |
| 626 | MultipleFailureDetectionError | Found user transaction declined by issuer. Try again later! |
| 627 | CSDetectionError | The CrossSellingFilter blocks duplicated transactions when an approved transaction has been found on another mid. That is, if the transaction has already been processed successfully on a different mid and within the specified time frame, context entity / scope and possibly within the issuer scope (or not), it will be rejected in order to prevent duplicates. |
| 628 | RecurringLimitExceededError | Amount/count by recurring subscription exceeded. |
| 629 | IrisFilterDeclinedError | Transaction declined by risk management. |
| 630 | IrisFilterOnHoldError | Transaction on hold, a manual review will be done |
| 690 | AvsError | Address Verification failed. |
| 691 | MaxMindRiskError | If a transaction is considered high risk by MaxMind minFraud service, a MaxMindRiskError is raised. |
| 692 | ThreatMetrixRiskError | Transaction declined by ThreatMetrix risk module. |
| 693 | IpNotWhitelistedError | Transaction declined by risk management, IP is NOT whitelisted! |
| 694 | DomainBlacklistedError | Transaction declined by risk management, domain is blacklisted! |
| 695 | FraudError | Risk Error: Please contact the risk team! |
| 696 | IbanBlacklistError | Transaction declined by risk management, iban blacklisted! |

**Tokenization Errors**

| Code | Name | Description |
|---|---|---|
| 701 | ConsumerUniquenessError | Consumer with this consumer_id, email combination already exists! |
| 702 | InvalidConsumerError | Consumer not found! |
| 703 | DisabledConsumerError | Consumer is disabled! |
| 700 | TokenizationError | General tokenization error. |
| 710 | TokenizationNotEnabledError | Tokenization is not enabled for the merchant or the terminal! Contact support. |
| 720 | InvalidTokenTypeError | Unsupported token type! |
| 730 | InvalidTokenError | Invalid token! |
| 740 | DetokenizeForbiddenError | Detokenize action is forbidden! |
| 741 | TokenizeForbiddenError | Tokenize action is forbidden! |
| 742 | UpdateTokenForbiddenError | Update token action is forbidden! |
| 743 | ValidateTokenForbiddenError | Validate token action is forbidden! |
| 744 | DeleteTokenForbiddenError | Delete token action is forbidden! |

**Client-side encryption-related errors**

750 | ClientSideEncryptionError | Client-side encryption error! 751 | ClientSideEncryptionNotEnabledError | Client-side Encryption is not enabled!

**KYC Errors**

| Code | Name | Description |
|---|---|---|
| 800 | KycServiceError | General KYC Service Error |

| | | |
|---|---|---|
| 801 | DocumentMimeTypeUnsupportedError | Uploaded document MIME type is not supported by KYC provider |
| 802 | InvalidRequestAttributesError | Passed attributes are invalid! |
| 803 | KycServiceNotConfiguredError | KYC Services not configured for Merchant! |
| 804 | KycServiceProviderError | KYC Service provider Error! |
| 805 | KycServiceNotificationError | Notification already received |
| 806 | KycServiceUnacceptableMerchantStateError | Merchant state does not allow using KYC Service API! |

**Remote Errors**

| Code | Name | Description |
|---|---|---|
| 900 | RemoteError | Some error occurred on the issuer. Contact support. |
| 910 | RemoteSystemError | Some error occurred on the issuer |
| 920 | RemoteConfigurationError | Issuer configuration error |
| 930 | RemoteDataError | Some passed data caused an error on the issuer |
| 940 | RemoteWorkflowError | Remote workflow error |
| 950 | RemoteTimeoutError | Issuer has timeouted with clearing network |

> Description can be slightly different for error messages per acquirer, but the error classes will be as documented.

# Client Integrations

There are client libraries and examples for a few programming languages to ease the merchant integration effort:

| Language | Github | Description |
|---|---|---|
| .NET | Genesis .NET | .NET client library |
| Java | Genesis Java | Java client library |
| Kotlin | Genesis Android | Android client library |
| Node.js | Genesis Node | Node.js client library |
| PHP | Genesis PHP | PHP client library |
| Swift | Genesis Swift | iOS client library |
| Ruby | Genesis Ruby | Ruby client library |

Should you have any questions or suggestions regarding the client libraries and improvements, contact the IT Support team at tech-support@e-comprocessing.com.

You can also fork the repo(s) and send us pull requests directly at our Github account.

# Client-side encryption

The *Client-side Encryption* (CSE) allows merchants to accept payments on their website while encrypting card data in their customer's browser with our JavaScript encryption library.

> The CSE must be used in combination with one of our classic Client Integrations.

To help merchants encrypt all sensitive card data on the customer's side, E-ComProcessing hosts the JavaScript library and merchant's encryption key. In addition, the merchants can decide to host the library by themselves, but we strongly recommend the Client-side encryption library hosted by our services to be used.

If the merchants want to use the Client-Side Encryption (CSE) library, it needs to contact tech-support@e-comprocessing.com in order to enable this feature for them. Then the merchant will be allowed to obtain the public key and the library in the merchant web console.

## Client side

```
<head>
  ...
  <script src="https://{CDN}/encrypto-{VERSION}.js"
          integrity="sha512-lJxHl93A/b8peqxz/mdLj7jD58N2zvHiiYhw8...=="
          crossorigin="anonymous"></script>
</head>

<body>
  <form method="POST" action="/request-payment" id="encrypto-form">
    <div>
      <input type="text" data-encrypted-name="card_number"/>
      <input type="text" data-encrypted-name="card_holder"/>
      ...
      <input type="text" data-encrypted-name="cvv"/>
      ...
      <input type="text" name="country"/>
      <input type="text" name="city"/>
    </div>
    ...
    <input type="submit" value="Pay"/>
  </form>

  <script>
    var publicKey = '...';

    Encrypto.createEncryptedForm(publicKey, {   // Required
      // formId: 'encrypto-form',               // Optional
      // onSubmit: function(form) {             // Optional
      //   console.log(form.fields);
      // }
    })
  </script>
</body>
```

First, the merchant needs to create a payment form integrated with the Client-Side Encryption (CSE) library which can be retrieved from the **Client Side Encryption** panel in the merchant console at the merchant configuration page.

Make sure that payment form contains all required fields for the transaction type which it's going to be used. Consult with our Transactions documentation. Don't forget to replace the form action with the payment handler URL of the merchant's server.

Flag all card input fields for encryption by annotating them with the `data-encrypted-name` attribute. The name attribute should not be used for card input fields. The fields allowed for encryption are `card_holder`, `card_number`, `expiration_month`, `expiration_year`, and `cvv`.

> Use "data-encrypted-name" attribute for card input fields. This technique protects the merchant's server from receiving unencrypted card data and avoids any impact on the transaction security and PCI regulations compliance.

Eventually, the form may have a custom ID attribute. The `formId` option can be used to set any string as an ID for the payment form. Make sure to update the HTML form with the configured option. They both must match.

**Options**

| option | type | description |
|---|---|---|
| formId | string | Use to set custom form ID. Default value `encrypto-form` |
| onSubmit | function | Use to set custom on submit callback |

**PREVENT FORM SUBMIT ACTION**

In the case of a single-page application or a form that uses AJAX, maybe it's not desirable the form to reload the page when the payment gets submitted. For that reason, we provide `onSubmit` option. This option gives access to all the form data(including the encrypted fields) and allows to submit it via any AJAX library.

**JAVASCRIPT ONLY**

```
<head>
  ...
  <script src="https://{CDN}/encrypto-{VERSION}.js"
          integrity="sha512-lJxHl93A/b8peqxz/mdLj7jD58N2zvHiiYhw8...=="
          crossorigin="anonymous"></script>
</head>

<body>
  <script>
    var publicKey = '...';
    var cse = Encrypto.createEncryption(publicKey);

    let data = {
      cvv: '123',
      card_number: '42000...'
    };

    var encryptedData = cse.encrypt(data);
  </script>
</body>
```

In case the merchant does not have an HTML form, our library provides HTML-independent encryption. In this scenario, it's important to remember it's the merchant responsibility to make sure the card data is encrypted before sending it to the server.

> Always encrypt the card data before sending to the merchant's server.

The JavaScript-only option can be convenient in case of a more complex single-page application which relies on state management library before sending any data to the server.

---

## Server side

From the merchant's server, an HTTP POST request needs to be made to the gateway API endpoints. The workflow is the same as in the classic Client Integrations. The only difference when the Client-Side Encryption (CSE) library is used is that our gateway will receive the card data encrypted.

> There is no need to change anything in the merchant's server if it's already using any of the Client Integrations.

Do not worry about the decryption. Our gateway will handle the API request as a standard transaction. Make sure to always use the correct public key in the client-side code.

---

# Shopping Carts

---

Genesis has a number of shopping cart plugins to ease the merchant integration effort:

| Shopping cart | Github | Marketplace | Description |
|---|---|---|---|
| Shopify | Internal integration | Internal integration | Shopping cart Integration for Shopify |

Should you have any questions or suggestions regarding the shopping cart plugins and improvements, contact the IT Support team at tech-support@e-comprocessing.com.

You can also fork the repo(s) and send us pull requests directly at our Github account.

---

# Testing

---

For information about the testing environment see Environments.

For testing first login to the gateway admin and create a terminal.

The url for test admin is:

`https://staging.merchant.e-comprocessing.net/`

The api base url for test processing is:

`https://staging.gate.e-comprocessing.net/process/TERMINAL_TOKEN`

The api base url for test single transaction reconciling is:

`https://staging.gate.e-comprocessing.net/reconcile/TERMINAL_TOKEN`

The api base url for test date range reconciling is:

`https://staging.gate.e-comprocessing.net/reconcile/by_date/TERMINAL_TOKEN`

For testing the gateway the following credit card numbers can be used:

| card number | card brand | transaction result |
|---|---|---|
| 4200000000000000 | Visa | successful transaction |
| 4111111111111111 | Visa | transaction declined |
| 5555555555554444 | Master Card | successful transaction |
| 5105105105105100 | Master Card | transaction declined |

For 3DSecure testing the following credit card numbers can be used:

## 3DS v2

For more specifics and flows regarding the 3DSv2 authentication protocol, go to 3DSv2 section.

| Scenario | 3DSecure Method | 3DSecure Challenge | Result | Card Number | Note |
|---|---|---|---|---|---|
| Frictionless | - | - | Authenticated | 4012000000060085 | |
| Frictionless | - | - | Authenticated | 4901170000000003 | |
| Frictionless | - | - | Authenticated | 4901164281364345 | |
| Frictionless | - | - | Authenticated | 4378510000000004 | |
| Frictionless | - | - | Attempted | 4456530000001005 | |
| Frictionless | - | - | Attempted | 5200000000001054 | |
| Frictionless | Y | - | Authenticated | 4066330000000004 | |
| Frictionless | Y | - | Authenticated | 5200000000001021 | |
| Low risk exemption accepted (MasterCard) | - | - | Authenticated | 5169750000001111 | Used only for synchronous 3DS workflow. |
| Low risk exemption accepted (Visa) | - | - | Authenticated | 4378510000000004 | Used only for synchronous 3DS workflow. |
| Frictionless | - | - | Not authenticated | 4111110000000922 | |
| Frictionless | - | - | Not authenticated | 5200000000001047 | |
| Frictionless | Y | - | Not authenticated | 4111112232423922 | |
| Challenge | - | Y | Choose Challenge result | 4918190000000002 | |
| Challenge | - | Y | Choose Challenge result | 5433300000000133 | |
| Challenge | Y | Y | Choose Challenge result | 4938730000000001 | |
| Challenge | Y | Y | Choose Challenge result | 5200000000001005 | |

In test mode, successful transaction XML responses will include the following error message: ”TESTMODE: No real money will be transferred!”

## AVS

The following amounts can be used to return specific avs response code and avs response description:

| Code | Amount | BIN Country | Response Description |
|---|---|---|---|
| A | 1.00 | UK, US | Address matches - ZIP Code does not match |
| B | 1.01 | UK, US | Street address match, Postal code in wrong format (international issuer) |
| C | 1.02 | UK, US | Street address and postal code in wrong formats |
| D | 1.03 | UK, US | Street address and postal code match (international issuer) |
| F | 1.04 | UK | Address does compare and five-digit ZIP code does compare (UK only) |
| G | 1.05 | UK, US | Service not supported by non-US issuer |
| I | 1.06 | UK, US | Address information not verified by international issuer |
| M | 1.07 | UK, US | Street Address and Postal code match (international issuer) |
| N | 1.08 | UK, US | No Match on Address (Street) or ZIP |
| P | 1.09 | UK, US | Postal codes match, Street address not verified due to incompatible formats |
| R | 1.10 | UK, US | Retry, System unavailable or Timed out |
| S | 1.11 | UK, US | Service not supported by issuer |
| U | 1.12 | UK, US | Address information is unavailable |
| W | 1.13 | US | 9-digit ZIP matches, Address (Street) does not match |
| X | 1.14 | US | Exact AVS Match |
| Y | 1.15 | US | Address (Street) and 5-digit ZIP match |
| Z | 1.16 | US | 5-digit ZIP matches, Address (Street) does not match |

Visa, Maestro and Mastercard card brands can be used for all avs response codes except for 1.14, which only works with Maestro and Mastercard.

The AVS response does not depend on the status of the transaction.

# Status Page

statuspage.io is a popular service allowing to track server status, infrastructure notifications, and others.

Note that you can sign up for E-ComProcessing's status page.

It allows to sign up via email or SMS or both, and receive notifications for our payment services and any planned maintenance windows, upgrades, or similar in the future should the need arise.

# Infrastructure and Uptime

Genesis is hosted in two data centers respectively in Berlin and Amsterdam, and features a state-of-the-art, active-active infrastructure setup. As such, it employs load balancing and failover on the DNS layer, and you should be using and requesting the API nodes and web apps only via their dedicated DNS names.

No hard-coding of IP addresses should be performed on the customers' systems, as this will prevent the customer to take advantage of the failover in case one of the data centers has issues or throughout maintenance windows and current processing happens through one data center only, however rare this might be. In addition, load balancing of the customers' volume is also impacted if IP addresses are hardcoded.

Finally, note that the DNS load balancing and failover layer has a TTL of 30 seconds, and will sense any issues returning the right IP addresses to use, for both API nodes and web apps alike, at all times.

As a highly available payment gateway platform, Genesis strives to achieve an uptime SLA of 99.99 percent on a yearly basis.

# Penetration Testing Warning

It is important that merchants read and understand the activities that are explicitly prohibited when using the payment gateway services.

While merchants are encouraged to perform best practice security testing on their own websites and applications, merchants must ensure under all circumstances that scans exclude the payment gateway Web Payment Form (WPF), Processing API, and merchant console.

| Action | prohibited? |
|---|---|
| Penetration testing of any gateway services | Prohibited |
| Load testing | Prohibited. During integration testing, ensure minimum number of requests |
| Exploiting common security vulnerabilities | Prohibited |
| Injecting malicious data | Prohibited |
| Bypassing validation and security checks | Prohibited |
| Subverting ACLs and user permissions | Prohibited |
| Port scanning and service discovery | Prohibited |
| Usage of ping/traceroute | Acceptable for short term debugging purposes |

Note that merchants that do not abide by the above policy will be immediately blacklisted, resulting in terminating access to the WPF, Processing API, and merchant console.

# AVS Status Codes

There are two cases for AVS codes, single and two-character codes.

**SINGLE CHARACTER CODE**

AVS code description is mentioned in the below table.

| Code | Descriptive response | AVS Response text |
|---|---|---|
| A | Approved | Postcode and address matched |
| B | Partial Approved | Postcode matched; address not checked |
| C | Partial Approved | Postcode matched; address not matched |
| D | Partial Approved | Address matched; postcode not checked |
| E | Not Sent To Acquirer | Postcode and address not checked |
| F | Partial Approved | Address matched; postcode not matched |
| G | Partial Approved | Postcode not checked; address not matched |
| H | Not Supplied By Shopper | Postcode and address not supplied by shopper/merchant |
| I | Partial Approved | Address not checked; postcode not matched |
| J | Failed | Postcode and address not matched |

**TWO-CHARACTERS CODE**

When sending a transaction for address verification or card verification, an AVS code will be returned.

The first character of the AVS code represents the entity that was responsible for generating that code. See below table for details.

| Code | Description |
|---|---|
| 2 | Response provided by Intermediate Processor |
| 5 | Response provided by Issuer Processor |

The second character of the AVS code description is mentioned in the below table.

| Code | Summary | VISA | MasterCard |
|---|---|---|---|
| A | Partial Match | Street address matches, but 5-digit and 9-digit postal code do not match. | Address matches, ZIP does not. |
| B | Partial Match (International Transaction) | Not applicable. | Street addresses match. Postal code not verified due to incompatible formats. (Acquirer sent both street address and postal code.) |
| C | No Match (International Transaction) | Not applicable. | Street address and postal code not verified due to incompatible formats. (Acquirer sent both street address and postal code.) |
| D | Full Match (International Transaction) | Not applicable. | Street addresses and postal codes match. |
| F | Full Match (UK only) | Not applicable. | Street address and postal code match. Applies to U.K. only. |

| | | | |
|---|---|---|---|
| G | Not Supported (International Transaction) | Not applicable. | Address information not verified for international transaction. |
| I | No Match (International Transaction) | Not applicable. | Address information not verified. |
| M | Full Match (International Transaction) | Not applicable. | Street address and postal code match. |
| N | No Match | No match. Acquirer sent postal/ZIP code only, or street address only, or both postal/ZIP and street address. | Neither address nor postal code matches. |
| P | Partial Match (International Transaction) | Not applicable. | Postal code match. Street address not verified because of incompatible formats. (Acquirer sent both street address and postal code.) |
| R | System Unavailable | Retry: System unavailable or timed out. Issuer ordinarily performs address verification but was unavailable. | Retry, system unable to process. |
| S | Not Supported | Not applicable. | AVS currently not supported. |
| U | System Unavailable | Address not verified for domestic transaction. Address not verified for international transaction. Issuer is not an AVS participant, or AVS data was present in the request but issuer did not return an AVS result, or V.I.P. performed address verification on behalf of the issuer and there was no address record on file for this account. | No data from issuer/Authorization System. Information not available. |
| W | Partial Match (US only) | Not applicable. | For U.S. addresses, nine-digit postal code matches, address does not; for address outside the U.S., postal code matches, address does not. |
| X | Full Match | Not applicable. | For U.S. addresses, nine-digit postal code and address matches; for addresses outside the U.S., postal code and address match. |
| Y | Full Match | Street address and postal/ZIP match. | For U.S. addresses, five-digit postal code and address matches. |
| Z | Partial Match | Postal/ZIP match, street addresses do not match or street address not inclued in request. | For U.S. addresses, five-digit postal code matches, address does not. |

AVS status codes are valid for cards issued from United States and United Kingdom. For other countries status message will be returned with error description.

# CVV Result Codes

| Code | Summary |
|---|---|
| M | Match |
| N | No match |
| S | Should be on card |
| P | Not processed |
| U | Issuer does not participate |

# Level 3 Travel Data

Level 3 travel data is supplied as optional data to the standard API request. If the supplied is valid travel data then the transaction will be processed as a travel transaction and will qualify for the travel Incentive rates. Otherwise the transaction will be processed normally as a regular transaction. Note that the travel data will be stored as part of the transaction in all cases.

Travel data is supported for Authorize, Authorize3D, Capture, Sale, Sale3D, Refund.

The following travel types are supported Airline Itinerary Data (AID), Car Rental, Hotel Rental, Ancillary Charges, Misc Charges.

Check the travel related transaction special cases.

## Travel Types

**AIRLINE ITINERARY DATA (AID)**

**Airline Ticket transaction With Airline Itinerary Data (AID)**

**MasterCard**

Contract Merchant Category Code must be 3000-3350, 4511, 4722, 4723 or 5962

Master Card

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <ticket>
       <ticket_number>123456789012345</ticket_number>
       <passenger_name>Test Example</passenger_name>
       <customer_code>1</customer_code>
       <issuing_carrier>AAAA</issuing_carrier>
       <total_fare>5000</total_fare>
       <agency_name>Agency</agency_name>
       <agency_code>AG001</agency_code>
     </ticket>
     <legs>
       <leg>
         <departure_date>2018-02-05</departure_date>
         <carrier_code>12</carrier_code>
         <service_class>1</service_class>
```

```
        <origin_city>VAR</origin_city>
        <destination_city>FRA</destination_city>
        <stopover_code>0</stopover_code>
        <fare_basis_code>1</fare_basis_code>
        <flight_number>W6666</flight_number>
        <departure_time>11:37</departure_time>
        <departure_time_segment>A</departure_time_segment>
      </leg>
    </legs>
  </travel>
</payment_transaction>'
```

Master Card Multiple Legs

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <.../>
  <travel>
    <ticket>
      <ticket_number>123456789012345</ticket_number>
      <passenger_name>Test Example</passenger_name>
      <customer_code>1</customer_code>
      <issuing_carrier>AAAA</issuing_carrier>
      <total_fare>5000</total_fare>
      <agency_name>Agency</agency_name>
      <agency_code>AG001</agency_code>
    </ticket>
    <legs>
      <leg>
        <departure_date>2018-02-05</departure_date>
        <carrier_code>12</carrier_code>
        <service_class>1</service_class>
        <origin_city>VAR</origin_city>
        <destination_city>FRA</destination_city>
        <stopover_code>0</stopover_code>
        <fare_basis_code>1</fare_basis_code>
        <flight_number>W6666</flight_number>
        <departure_time>11:37</departure_time>
        <departure_time_segment>A</departure_time_segment>
      </leg>
      <leg>
        <departure_date>2018-02-05</departure_date>
        <carrier_code>12</carrier_code>
        <service_class>1</service_class>
        <origin_city>VAR</origin_city>
        <destination_city>BER</destination_city>
        <stopover_code>0</stopover_code>
        <fare_basis_code>1</fare_basis_code>
        <flight_number>W6666</flight_number>
        <departure_time>11:37</departure_time>
        <departure_time_segment>A</departure_time_segment>
      </leg>
    </legs>
  </travel>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **ticket** | required* | | |
| ticket_number | required* | String(15) | The number on the ticket. |
| passenger_name | required* | String(29) | The name of the passenger. May be the cardholder name if the passenger name is unavailable. Must not be blank. |
| customer_code | required* | String(17) | The customer code. Internal Reference. |
| issuing_carrier | optional | String(4) | Contains the standard abbreviation for the airline or railway carrier issuing the ticket. |
| total_fare | required* | Integer | Total amount of the ticket and should equal the amount of the transaction. |
| agency_name | optional | String(30) | An entry should be supplied if a travel agency issued the ticket. |
| agency_code | optional | String(8) | An entry should be supplied if a travel agency issued the ticket. |
| confirmation_information | required* | String(474) | Confirmation Information |
| date_of_issue | required* | String(10) | Date Of Issue |
| **legs** | required* | | Max legs 10 |
| **leg** | required* | | |
| departure_date | required | String(10) | The departure date. Date can be in future. |
| arrival_date | required* | String(10) | The arrival date. Date can be in future. |
| carrier_code | required* | String(2) | Contains the standard abbreviation for the airline or railway carrier issuing the ticket. This should not contain all spaces or zeros. Code indicating name of carrier. |
| service_class | required* | String(1) | The service type. i.e. Coach, First Class. Required for reduced interchange. |
| origin_city | required* | String(3) | The originating airport name’s standard abbreviation. This should not contain all spaces or zeroes. |
| destination_city | required* | String(3) | The destination airport or railway name’s standard abbreviation. |
| stopover_code | required* | String(1) | A code indicating whether there was a direct or a non-direct flight or route on the same ticket number. Allowed values: 0, 1 |
| fare_basis_code | optional | String(6) | A code that carriers assign to a particular ticket type, such as business class or discounted/ non-re fundable. |
| flight_number | optional | String(5) | The number that the operating or marketing carrier assigned. |
| departure_time | optional | String(5) | The time of departure provided by the airline or railway, per trip leg. |
| departure_time_segment | optional | String(1) | Departure Time Segment. Allowed values: A, P |
| **taxes** | optional | | Max taxes 10 |
| **tax** | optional | | |
| fee_amount | required* | Integer | Fee Amount |
| fee_type | required* | String(8) | Fee Type |

required* = conditionally required

**Visa**

Contract Merchant Category Code must be 3000-3350, 4511, 4722, 4723 or 5962

Visa

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <ticket>
       <ticket_number>12345</ticket_number>
       <passenger_name>Emil Example</passenger_name>
       <customer_code>1</customer_code>
       <restricted_ticket_indicator>1</restricted_ticket_indicator>
       <agency_name>Agency</agency_name>
       <agency_code>AG001</agency_code>
       <confirmation_information>Confirmation</confirmation_information>
       <date_of_issue>2018-02-01</date_of_issue>
     </ticket>
     <legs>
       <leg>
         <departure_date>2018-02-01</departure_date>
         <carrier_code>2</carrier_code>
         <service_class>3</service_class>
         <origin_city>SOF</origin_city>
         <destination_city>VAR</destination_city>
         <stopover_code>0</stopover_code>
         <fare_basis_code>1</fare_basis_code>
         <flight_number>W6666</flight_number>
       </leg>
     </legs>
     <taxes>
       <taxe>
         <fee_amount>1000</fee_amount>
         <fee_type>Airport tax</fee_type>
       </taxe>
     </taxes>
   </travel>
 </payment_transaction>'
```

Visa Multiple Legs

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <ticket>
       <ticket_number>123456789012345</ticket_number>
       <passenger_name>Test Example</passenger_name>
       <customer_code>1</customer_code>
       <restricted_ticket_indicator>1</restricted_ticket_indicator>
       <agency_name>Agency</agency_name>
       <agency_code>AG001</agency_code>
     </ticket>
     <legs>
       <leg>
         <departure_date>2018-02-01</departure_date>
         <carrier_code>2</carrier_code>
         <service_class>3</service_class>
         <origin_city>SOF</origin_city>
         <destination_city>VAR</destination_city>
         <stopover_code>0</stopover_code>
         <fare_basis_code>1</fare_basis_code>
         <flight_number>W6666</flight_number>
       </leg>
       <leg>
         <departure_date>2018-02-01</departure_date>
         <carrier_code>2</carrier_code>
         <service_class>3</service_class>
         <origin_city>VAR</origin_city>
         <destination_city>FRA</destination_city>
         <stopover_code>0</stopover_code>
         <fare_basis_code>1</fare_basis_code>
         <flight_number>W6366</flight_number>
       </leg>
     </legs>
   </travel>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **ticket** | required* | | |
| ticket_number | required* | String(15) | The number on the ticket. |
| passenger_name | required* | String(29) | The name of the passenger. May be the cardholder name if the passenger name is unavailable. Must not be blank. |
| customer_code | required* | String(17) | The customer code. Internal Reference. |
| confirmation_information | required* | String(474) | Confirmation Information |
| date_of_issue | required* | String(10) | Date Of Issue |
| restricted_ticket_indicator | optional | String(1) | Space or 0 = No restriction, 1 = Restriction; Allowed values: Empty String, 0, 1 |
| agency_name | optional | String(30) | An entry should be supplied if a travel agency issued the ticket. |
| agency_code | optional | String(8) | An entry should be supplied if a travel agency issued the ticket. |
| **legs** | required* | | Max legs 10 |
| **leg** | required* | | |
| departure_date | required | String(10) | The departure date. Date can be in future. |

| | | | |
|---|---|---|---|
| arrival_date | required* | String(10) | The arrival date. Date can be in future. |
| origin_city | required* | String(3) | The originating airport name's standard abbreviation. This should not contain all spaces or zeroes. |
| carrier_code | required* | String(2) | Contains the standard abbreviation for the airline or railway carrier issuing the ticket. This should not contain all spaces or zeros. Code indicating name of carrier. |
| service_class | required* | String(1) | The service type. i.e. Coach, First Class. Required for reduced interchange. |
| stopover_code | required* | String(1) | A code indicating whether there was a direct or a non-direct flight or route on the same ticket number. Allowed values: 0, 1 |
| destination_city | required* | String(3) | The destination airport or railway name's standard abbreviation. |
| fare_basis_code | optional | String(6) | A code that carriers assign to a particular ticket type, such as business class or discounted/ non-re fundable. |
| flight_number | optional | String(5) | The number that the operating or marketing carrier assigned. |
| **taxes** | optional | | Max taxes 10 |
| **tax** | optional | | |
| fee_amount | required* | Integer | Fee Amount |
| fee_type | required* | String(8) | Fee Type |

`required* = conditionally required`

**CAR RENTAL**

**MasterCard**

Contract Merchant Category Code must be 3351-3500, 4722, 4723, 5962, 7512, 7513, 7519

Master Card

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <rentals>
       <car_rental>
         <purchase_identifier>12478</purchase_identifier>
         <class_id>3</class_id>
         <pickup_date>2018-02-05</pickup_date>
         <renter_name>Emil Example</renter_name>
         <return_city>Varna</return_city>
         <return_state>VAR</return_state>
         <return_country>BGR</return_country>
         <return_date>2018-02-06</return_date>
         <renter_return_location_id>12478</renter_return_location_id>
         <customer_code>1</customer_code>
       </car_rental>
     </rentals>
   </travel>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **rentals** | required | | |
| **car_rental** | required | | |
| purchase_identifier | required* | String(9) | Rental Agreement Number / Hotel Folio Number. |
| class_id | required* | String(4) | The car rental classification. Allowed values: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 9999 |
| pickup_date | required | String(10) | Car rental Pick-up date. |
| renter_name | required* | String(20) | The Renter Name |
| return_city | required* | String(18) | The Rental Return City |
| return_state | required* | String(3) | The Rental Return State |
| return_country | required* | String(3) | The Rental Return Country |
| return_date | required | String(10) | Car Rental return date |
| renter_return_location_id | required* | String(10) | Expenses or Car Rental code, Address, phone number, etc. Identifying Rental Return Location. |
| customer_code | required* | String(17) | The customer code. Internal Reference. |

`required* = conditionally required`

**Visa**

Contract Merchant Category Code must be 3351-3500, 4722, 4723, 5962, 7512, 7513, 7519

Visa

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <rentals>
       <car_rental>
         <purchase_identifier>12478</purchase_identifier>
         <class_id>3</class_id>
         <pickup_date>2018-02-05</pickup_date>
         <renter_name>Emil Example</renter_name>
         <return_city>Varna</return_city>
         <return_state>VAR</return_state>
         <return_country>BGR</return_country>
         <return_date>2018-02-06</return_date>
         <renter_return_location_id>12478</renter_return_location_id>
         <customer_code>1</customer_code>
```

```
      </car_rental>
    </rentals>
  </travel>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **rentals** | required | | |
| **car_rental** | required | | |
| purchase_identifier | optional | String(25) | Rental Agreement Number / Hotel Folio Number |
| pickup_date | required* | String(10) | Car rental Pick-up date. |
| return_date | required* | String(10) | Car rental Return date. |
| extra_charges | optional | Array(6) | Additional charges added to customer bill after check-out. Each position can be used to indicate a type of charge; Allowed values: 1, 2, 3, 4, 5 |
| no_show_indicator | optional | String(1) | No show indicator; Allowed values: 0, 1 |

`required* = conditionally required`

**HOTEL RENTAL**

**MasterCard**

Contract Merchant Category Code must be 3501-3999, 4722, 4723, 5962, 7011

Master Card

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <.../>
  <travel>
    <rentals>
      <hotel_rental>
        <purchase_identifier>12478</purchase_identifier>
        <arrival_date>3</arrival_date>
        <departure_date>2018-02-05</departure_date>
        <customer_code>1</customer_code>
      </hotel_rental>
    </rentals>
  </travel>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **rentals** | required | | |
| **hotel_rental** | required | | |
| purchase_identifier | required* | String(10) | Rental Agreement Number / Hotel Folio Number. |
| arrival_date | required | String(10) | Hotel check-in date. |
| departure_date | required | String(10) | The departure date. Date can be in future. |
| customer_code | required* | String(17) | The customer code. Internal Reference. |

`required* = conditionally required`

**Visa**

Contract Merchant Category Code must be 3501-3999, 4722, 4723, 5962, 7011

Visa

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <.../>
  <travel>
    <rentals>
      <hotel_rental>
        <purchase_identifier>2</purchase_identifier>
        <arrival_date>2018-02-01</arrival_date>
        <extra_charges>467</extra_charges>
        <no_show_indicator>1</no_show_indicator>
      </hotel_rental>
    </rentals>
  </travel>
</payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **rentals** | required | | |
| **hotel_rental** | required | | |
| purchase_identifier | optional | String(25) | Rental Agreement Number / Hotel Folio Number |
| arrival_date | required* | String(10) | Hotel rental Pick-up date. |
| departure_date | required* | String(10) | Hotel rental Departure date. |

| | | | |
|---|---|---|---|
| extra_charges | optional | Array(6) | Additional charges added to customer bill after check-out. Each position can be used to indicate a type of charge. Allowed values: 2, 3, 4, 5, 6, 7 |
| no_show_indicator | optional | String(1) | No show indicator; Allowed values: 0, 1 |

`required* = conditionally required`

**ANCILLARY CHARGES**

**Ancillary Charges**

Charges/fees related to the ticket. These transactions are processed on a separate transaction, referenced to Airline transaction with AID.

**MasterCard**

Used to identify only Baggage Charges.

Contract Merchant Category Code must be 3000-3350, 4511, 4722, 4723 or 5962

Master Card

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <ticket>
       <ticket_reference_id>8b7e3575e5605ea7e1895707a3e92837</ticket_reference_id>
     </ticket>
     <charges>
       <charge>
         <type>BG</type>
       </charge>
     </charges>
   </travel>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **ticket** | required | | |
| ticket_reference_id | required | String(32) | Unique id of the ticket transaction |
| **charges** | required | | |
| **charge** | required | | |
| type | required | String(2) | This field will contain the Service Category Code for the primary type of service that has been provided. Allowed values: BG |

`required* = conditionally required`

**Visa**

Used to identify charges for a number of ancillary services such as ticket upgrades, baggage fee, food & beverage purchases which are not purchased as part of the original ticket. Also used for charges/fees related to partial airline ticket refunds or ticket cancellations.

Contract Merchant Category Code must be 3000-3350, 4511, 4722, 4723 or 5962

Visa

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <ticket>
       <ticket_reference_id>8b7e3575e5605ea7e1895707a3e92837</ticket_reference_id>
       <ticket_document_number>1111</ticket_document_number>
       <issued_with_ticket_number>12321</issued_with_ticket_number>
     </ticket>
     <charges>
       <charge>
         <type>BF</type>
         <sub_type>BG</sub_type>
       </charge>
     </charges>
   </travel>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **ticket** | required | | |
| ticket_reference_id | required | String(32) | Unique id of the ticket transaction |
| ticket_document_number | required | String(15) | This field will contain the form number assigned by the carrier for the transaction. |
| issued_with_ticket_number | required | String(15) | If this purchase has a connection or relationship to another purchase, such as baggage fee for a passenger transport ticket, this field must contain the document number for the other purchase. |
| **charges** | required | | |
| **charge** | required | | |
| type | required | String(2) | This field will contain the Service Category Code for the primary type of service that has been provided. Allowed values: BF, BG, CF, CG, CO, FF, GF, GT, IE, LG, MD, ML, OT, PA, PT, SA, SB, SF, ST, TS, UN, UP, WI |
| sub_type | required | String(2) | This field will contain the Service Category Code for the secondary type of service that has been provided Allowed values: BF, BG, CF, CG, CO, FF, GF, GT, IE, LG, MD, ML, OT, PA, PT, SA, SB, SF, ST, TS, UN, UP, WI |

`required* = conditionally required`

**MISCELLANEOUS CHARGES**

Miscellaneous charges related to the travel, but not related to the ticket.

**MasterCard**

Contract Merchant Category Code must be 3000-3350, 4511, 4722, 4723 or 5962

Master Card

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <ticket>
       <ticket_reference_id>8b7e3575e5605ea7e1895707a3e92837</ticket_reference_id>
     </ticket>
     <charges>
       <charge>
         <type>MISC</type>
       </charge>
     </charges>
   </travel>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **ticket** | required | | |
| ticket_reference_id | required | String(32) | Unique id of the ticket transaction |
| **charges** | required | | |
| **charge** | required | | |
| type | required | String(4) | This field will contain the Service Category Code for the primary type of service that has been provided. Allowed values: MISC |

`required* = conditionally required`

**Visa**

Contract Merchant Category Code must be 3000-3350, 4511, 4722, 4723 or 5962

Visa

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <ticket>
       <ticket_reference_id>8b7e3575e5605ea7e1895707a3e92837</ticket_reference_id>
     </ticket>
     <charges>
       <charge>
         <type>MISC</type>
       </charge>
     </charges>
   </travel>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | required | | |
| **ticket** | required | | |
| ticket_reference_id | required | String(32) | Unique id of the ticket transaction |
| **charges** | required | | |
| **charge** | required | | |
| type | required | String(4) | This field will contain the Service Category Code for the primary type of service that has been provided. Allowed values: MISC |

`required* = conditionally required`

---

# Special Cases

**TRAVEL AUTHORIZE (3D) AND CAPTURE**

The Capture travel data is always merged with the Authorize travel data and overrides the Authorization fields (where they are present in both transactions) before validating. This makes the required travel data for Authorizations optional. This logic is applied for all Travel Types. Because of this, there are 4 scenarios for submitting travel Authorization and Capture.

**Travel Authorize (3D) and Travel Capture**

In this scenario the Authorize and Capture transaction requests are submitted with valid travel data. The travel data that will be used for the transaction processing is the data submitted with the Capture.

Example Travel Authorize

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>authorize</transaction_type>
   <transaction_id>1915c52b28ff29edb12bc617ac102a03</transaction_id>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
```

```
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4200000000000000</card_number>
  <cvv>834</cvv>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <travel>
    <ticket>
      <ticket_number>12345678123456</ticket_number>
      <passenger_name>Passenger 01</passenger_name>
      <customer_code>123</customer_code>
      <restricted_ticket_indicator>0</restricted_ticket_indicator>
      <agency_name>Agency</agency_name>
      <agency_code>AG001</agency_code>
    </ticket>
    <legs>
      <leg>
        <departure_date>2017-03-10</departure_date>
        <carrier_code>VX</carrier_code>
        <service_class>J</service_class>
        <origin_city>DUB</origin_city>
        <destination_city>ATL</destination_city>
        <stopover_code>1</stopover_code>
        <fare_basis_code>0</fare_basis_code>
        <flight_number/>
      </leg>
    </legs>
  </travel>
</payment_transaction>'
```

Example Travel Capture

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>capture</transaction_type>
  <transaction_id>7fcc6097153beeec1dd1e71f643198d8</transaction_id>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <reference_id>1f2ee425e7c3159c60743098071771eb</reference_id>
  <travel>
    <ticket>
      <ticket_number>12345678123456</ticket_number>
      <passenger_name>Passenger 01</passenger_name>
      <customer_code/>
      <restricted_ticket_indicator>0</restricted_ticket_indicator>
      <agency_name>New Agency</agency_name>
      <agency_code>AGN001</agency_code>
    </ticket>
    <legs>
      <leg>
        <departure_date>2017-03-10</departure_date>
        <carrier_code>VX</carrier_code>
        <service_class>J</service_class>
        <origin_city>DUB</origin_city>
        <destination_city>ATL</destination_city>
        <stopover_code>1</stopover_code>
        <fare_basis_code>0</fare_basis_code>
        <flight_number/>
      </leg>
    </legs>
  </travel>
</payment_transaction>'
```

**Non Travel Authorize (3D) and Travel Capture**

In this scenario the Authorize request doesn’t contain the travel data. Valid travel data is submitted in the Capture transaction request. The travel data that will be used for the transaction processing is the data submitted with the Capture.

Example Non Travel Authorize

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>authorize</transaction_type>
  <transaction_id>1915c52b28ff29edb12bc617ac102a03</transaction_id>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4200000000000000</card_number>
  <cvv>834</cvv>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
</payment_transaction>'
```

Example Travel Capture

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>capture</transaction_type>
  <transaction_id>7fcc6097153beeec1dd1e71f643198d8</transaction_id>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <reference_id>1f2ee425e7c3159c60743098071771eb</reference_id>
  <travel>
    <ticket>
      <ticket_number>12345678123456</ticket_number>
      <passenger_name>Passenger 01</passenger_name>
      <customer_code/>
      <restricted_ticket_indicator>0</restricted_ticket_indicator>
      <agency_name>New Agency</agency_name>
      <agency_code>AGN001</agency_code>
    </ticket>
    <legs>
      <leg>
        <departure_date>2017-03-10</departure_date>
```

```
        <carrier_code>VX</carrier_code>
        <service_class>J</service_class>
        <origin_city>DUB</origin_city>
        <destination_city>ATL</destination_city>
        <stopover_code>1</stopover_code>
        <fare_basis_code>0</fare_basis_code>
        <flight_number/>
      </leg>
    </legs>
  </travel>
</payment_transaction>'
```

**Travel Authorize (3D) and Non Travel Capture**

In this scenario the Authorize request contains valid travel data. Travel data isn't submitted in the Capture transaction request. In this case the Capture transaction will inherit the travel data from the Authorize transaction. The travel data that will be used for the transaction processing is the data submitted with the Authorize.

Example Travel Authorize

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>authorize</transaction_type>
  <transaction_id>1915c52b28ff29edb12bc617ac102a03</transaction_id>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4200000000000000</card_number>
  <cvv>834</cvv>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <travel>
    <ticket>
      <ticket_number>1234567812345 6</ticket_number>
      <passenger_name>Passenger 01</passenger_name>
      <customer_code>123</customer_code>
      <restricted_ticket_indicator>0</restricted_ticket_indicator>
      <agency_name>Agency</agency_name>
      <agency_code>AG001</agency_code>
    </ticket>
    <legs>
      <leg>
        <departure_date>2017-03-10</departure_date>
        <carrier_code>VX</carrier_code>
        <service_class>J</service_class>
        <origin_city>DUB</origin_city>
        <destination_city>ATL</destination_city>
        <stopover_code>1</stopover_code>
        <fare_basis_code>0</fare_basis_code>
        <flight_number/>
      </leg>
    </legs>
  </travel>
</payment_transaction>'
```

Example Non Travel Capture

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>capture</transaction_type>
  <transaction_id>7fcc6097153beeec1dd1e71f643198d8</transaction_id>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <reference_id>1f2ee425e7c3159c60743098071771eb</reference_id>
</payment_transaction>'
```

**Partial Travel Authorize (3D) and Partial Travel Capture**

In this scenario the Authorize request contains only part of the travel data. The Capture transaction request contains also part of the travel data. The travel data from the Authorize will be merged with the Capture travel data. The Capture travel data will complete/override the travel fields in the Authorize. This merged data will be stored as Capture travel data. If the merged data is valid travel data then the transaction will be processed as travel using the travel data stored in the Capture. Otherwise the transaction will be processed as a regular Capture transaction.

Example Partial Authorize

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
<?xml version="1.0" encoding="UTF-8"?>
<payment_transaction>
  <transaction_type>authorize</transaction_type>
  <transaction_id>1915c52b28ff29edb12bc617ac102a03</transaction_id>
  <remote_ip>245.253.2.12</remote_ip>
  <amount>100</amount>
  <currency>USD</currency>
  <card_holder>Travis Pastrana</card_holder>
  <card_number>4200000000000000</card_number>
  <cvv>834</cvv>
  <expiration_month>12</expiration_month>
  <expiration_year>2026</expiration_year>
  <travel>
    <ticket/>
    <legs>
      <leg>
        <departure_date>2017-03-10</departure_date>
        <carrier_code>VX</carrier_code>
        <service_class>J</service_class>
        <origin_city>DUB</origin_city>
        <destination_city>ATL</destination_city>
        <stopover_code>1</stopover_code>
        <fare_basis_code>0</fare_basis_code>
        <flight_number/>
      </leg>
    </legs>
  </travel>
</payment_transaction>'
```

Example Partial Capture

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_type>capture</transaction_type>
   <transaction_id>7fcc6097153beeec1dd1e71f643198d8</transaction_id>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <reference_id>1f2ee425e7c3159c60743098071771eb</reference_id>
   <travel>
     <ticket>
       <ticket_number>12345678123456</ticket_number>
       <passenger_name>Passenger 01</passenger_name>
       <customer_code/>
       <restricted_ticket_indicator>0</restricted_ticket_indicator>
       <agency_name>New Agency</agency_name>
       <agency_code>AGN001</agency_code>
     </ticket>
     <legs/>
   </travel>
 </payment_transaction>'
```

**VISA REFUND**

The additional visa refund request parameters are applicable only when the reference transaction is Airline Itinerary Data (AID) or Ancillary Charges.

Visa Refund

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <.../>
   <travel>
     <ticket>
       <credit_reason_indicator_1>C</credit_reason_indicator_1>
       <credit_reason_indicator_2>A</credit_reason_indicator_2>
       <ticket_change_indicator>B</ticket_change_indicator>
     </ticket>
   </travel>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **travel** | optional | | |
| **ticket** | optional | | |
| credit_reason_indicator_1 | optional | String(1) | This field indicates the reason for a credit to the cardholder. Allowed values: A, B, P, O |
| credit_reason_indicator_2 | optional | String(1) | This field indicates the reason for a credit to the cardholder. Allowed values: A, B, P, O |
| ticket_change_indicator | optional | String(1) | This field will contain either a space or a code to indicate why a ticket was changed. Allowed values: C, N |

`required* = conditionally required`

---

# Allowed Values

**CAR RENTAL CLASSES**

| Value | Description |
|---|---|
| 1 | Mini |
| 2 | Subcompact |
| 3 | Economy |
| 4 | Compact |
| 5 | Midsize |
| 6 | Intermediate |
| 7 | Standard |
| 8 | Full size |
| 9 | Luxury |
| 10 | Premium |
| 11 | Minivan |
| 12 | 12 passenger van |
| 13 | Moving van |
| 14 | 15 passenger van |
| 15 | Cargo van |
| 16 | 12 foot truck |
| 17 | 20 foot truck |
| 18 | 24 foot truck |
| 19 | 26 foot truck |
| 20 | Moped |
| 21 | Stretch limousine |

| | |
|---|---|
| 22 | Regular limousine |
| 23 | Unique limousine |
| 24 | Exotic limousine |
| 25 | Small/medium truck |
| 26 | Large truck |
| 27 | Small SUV |
| 28 | Medium SUV |
| 29 | Large SUV |
| 30 | Exotic SUV |
| 9999 | Miscellaneous |

**CHARGE TYPES**

| Value | Description |
|---|---|
| BF | Bundled Service |
| BG | Baggage Fee |
| CF | Change Fee |
| CG | Cargo |
| CO | Carbon Offset |
| FF | Frequent Flyer |
| GF | Gift Card |
| GT | Ground Transport |
| IE | In-flight Entertainment |
| LG | Lounge |
| MD | Medical |
| ML | Meal / Beverage |
| OT | Other |
| PA | Passenger Assist Fee |
| PT | Pets |
| SA | Seat Fees |
| SB | Standby |
| SF | Service Fee |
| ST | Store |
| TS | Travel Service |
| UN | Unaccompanied Travel |
| UP | Upgrades |
| WI | Wi-Fi |
| MISC | Miscellaneous Airline Charges |

**CAR RENTAL EXTRA CHARGES**

| Value | Description |
|---|---|
| 1 | Gas |
| 2 | Extra Mileage |
| 3 | Late Return |
| 4 | 1 Way Ser Fee |
| 5 | Parking Violation |

**HOTEL RENTAL EXTRA CHARGES**

| Value | Description |
|---|---|
| 2 | Restaurant |
| 3 | Gift Shop |
| 4 | Mini Bar |
| 5 | Telephone |
| 6 | Laundry |
| 7 | Other |

**TICKET CHANGE INDICATORS**

| Value | Description |
|---|---|
| C | Change to existing Ticket |
| N | New ticket |

**CREDIT REASON INDICATORS**

| Value | Description |
|---|---|
| A | Passenger Transport Ancillary Cancellation |
| B | Travel Ticket and Passenger Transport |

| | |
|---|---|
| P | Partial Refund of Travel Ticket |
| O | Other |

# Genesis SCA Services

## General Info

Genesis SCA(Strong Customer Authentication) Services provides the ability to check if a transaction is in the scope of SCA. The API is synchronous and is based on RESTFul practices. Be sure to set `Content-type: application/json` in your headers.

To interact with the SCA API, you need to provide login credentials using standard HTTP Basic Authentication. (Credentials can be found in your Admin interface.)

## SCA Checker

This call is used to check if SCA is required

`POST /v1/sca/checker/:terminal_token`

> **recurring** field is deprecated and soon will be removed from list of supported fields. Please use **recurring_type** instead. In case both **recurring** and **recurring_type** are used together within request, the value submitted in the **recurring** will be ignored.

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/sca/checker/:terminal_token \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "card_number": "4200000000000000",
   "moto": false,
   "mit": false,
   "recurring": false,
   "transaction_amount": 350000,
   "transaction_currency": "EUR",
   "transaction_exemption": ""
 }'
```

Successful Response

Response

```
[
  {
    "sca_required": "no",
    "sca_result_reason": "Issuer out of the EEA",
    "exemption": null
  }
]
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| card_number | required | string(16) | Full card number or first 6 digits. |
| transaction_amount | required | number | Amount of transaction in minor currency unit. |
| transaction_currency | required | string(3) | Transaction currency |
| moto | optional | boolean | Signifies whether a MOTO (mail order telephone order) transaction is performed. |
| mit | optional | boolean | Signifies whether a MIT (merchant initiated transaction) is performed. |
| recurring | optional | boolean | Signifies whether subsequent recurring transaction is performed. NOTE: field is DEPRECATED |
| recurring_type | optional | string(255) | Specifies recurring type of the transaction: initial or subsequent. |
| transaction_exemption | optional | string(30) | Exemption |

`required* = conditionally required`

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| sca_required | string | Sca Required. Possible values are `yes`, `possible_exemption` or `no` |
| sca_result_reason | string | The reason for the returned SCA required |
| exemption | string | Detected exemption. Check SCA exemption values. |

**SCA EXEMPTION VALUES**

> `delegated_authentication` is currently not supported.

| Value |
|---|
| low_value |
| low_risk |
| trusted_merchant |
| corporate_payment |
| delegated_authentication |
| auth_network_outage |

**SCA REASON FOR NOT HONORING EXEMPTION VALUES**

| Value | Translation |
|---|---|

| | |
|---|---|
| 8901 | Merchant not participating in Visa trusted listing |
| 8902 | Issuer not participating in trusted listing program |
| 8903 | Cardholder has not trusted the merchant |
| 8904 | Response from issuer is indeterminate |
| 8905 | No entry for VMID was found in supplementary database |
| 8906 | TRA risk analysis did not meet exemption criteria |
| 8473 | Cardholder has not trusted the merchant (issuer supplied) |
| 8474 | Did not meet the exemption criteria (issuer supplied) trusted listing |
| 8A01 | Merchant not participating in delegated authentication |
| 8A02 | Issuer not participating in delegated authentication |
| 8A04 | Indeterminate or invalid issuer response |
| 8A06 | Did not meet exemption criteria |
| 8A07 | VMID invalid for service |
| 8A08 | CAVV Invalid value |
| 8A76 | Did not meet exemption criteria |

**SCA EXEMPTION RESULT VALUES**

| Value | Translation |
|---|---|
| 13 | Low value exemption not honoured |
| 22 | TRA exemption honoured |
| 23 | TRA exemption not honoured |
| 32 | Trusted merchant exemption honoured |
| 33 | Trusted merchant exemption not honoured |
| 42 | Secure corporate payment exemption honoured |
| 43 | Secure corporate payment exemption not honoured |
| 52 | Delegated authentication honoured |
| 53 | Delegated authentication not honoured |

# Funding Transactions

## General Info

Funding transactions are special types of transactions and they require identifier type and receiver related details including receiver name, country, account number and account type. See the below tables for available identifier types and receiver account types.

## Funding Transaction identifier types

| Value | Required MCCs |
|---|---|
| general_person_to_person | 4829, 6538, 6540 |
| person_to_person_card_account | 4829 |
| own_account | 4829, 6051, 6211, 6538, 6540, 7800, 7801, 7802, 7994, 7995, 9406 |
| own_credit_card_bill | 4829, 6538 |
| business_disbursement | any |
| government_or_non_profit_disbursement | any |
| rapid_merchant_settlement | any |
| general_business_to_business | any |
| own_staged_digital_wallet_account | any |
| own_debit_or_prepaid_card_account | any |

## Funding Transaction Business Application Identifier Types

| Value |
|---|
| funds_disbursement |
| pension_disbursement |
| account_to_account |
| bank_initiated |
| fund_transfer |
| person_to_person |
| prepaid_card_load |
| wallet_transfer |

| |
|---|
| liquid_assets |

## Funding Transaction Receiver Account types

| Value | Description |
|---|---|
| rtn_and_bank_account_number | Routing Transit Number and Bank Account Number |
| iban | International Bank Account Number |
| card_account | Card Account |
| email | Electronic Mail |
| phone_number | Phone Number |
| bank_account_number_and_bic | Bank Account Number and Business Identifier Code |
| wallet_id | Wallet ID |
| social_network_id | Unique Identifier for Social Network Application |
| other | Any other type |

# Account Name Inquiry

## Introduction

Account Name Inquiry is a functionality that enables an account cardholder's name to be checked against the name held by their issuing bank. The check is carried out in advance of a transaction at time of customer onboarding, just before a transaction, periodically, or on an ad-hoc basis.

The name match result is returned to the requester, or originator, consisting of match results (match/partial/no) for each name sent (first/middle/last), as well as an overall match result for the complete name set.

This service is available with Visa cards in Canada, USA, Europe, UK, Latin America and the Caribbean regions.

We recommend usage in account verification or zero-amount transactions. May lead to unexpected behavior in other cases.

## How to use account name inquiry parameters

**REQUESTS**

Merchants can send account name inquiry params in the request when creating Authorize, Authorize3d, Sale, Sale3d, InitRecurringSale, InitRecurringSale3d, AccountVerification, Payout or Credit transactions via the Processing API and/or WPF API.

Account Name Inquiry

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/process/TERMINAL-TOKEN \
  -X POST \
  -H "Content-Type: text/xml" \
  -d '
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_transaction>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
   <remote_ip>245.253.2.12</remote_ip>
   <amount>100</amount>
   <currency>USD</currency>
   <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
   <card_holder>Travis Pastrana</card_holder>
   <cvv>834</cvv>
   <expiration_month>12</expiration_month>
   <expiration_year>2026</expiration_year>
   <customer_email>travis@example.com</customer_email>
   <customer_phone>+1987987987987</customer_phone>
   <account_owner>
     <first_name>Travis</first_name>
     <middle_name>Joe</middle_name>
     <last_name>Pastrana</last_name>
   </account_owner>
 </payment_transaction>'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| **account_owner** | optional | | Account owner parameters related to account owner inquiry requests |
| first_name | optional | String(35) | Account owner's first name. |
| middle_name | optional | String(35) | Account owner's middle name. |
| last_name | optional | String(35) | Account owner's last name. |

`required* = conditionally required`

**RESPONSES**

**EXPECTED RESPONSE VALUES FOR THE `*_NAME` ATTRIBUTES ARE *M* - MATCH, *P* - PARTIAL MATCH OR *N* - NO MATCH**

**EXPECTED RESPONSE VALUES FOR THE `TYPE` ATTRIBUTE ARE *P* - PRIMARY OR *S* - SECONDARY.**

The matched text will be returned in the response if there is a response code match returned by the cardholder's issuing bank.

Successful Response

Response

```
 <?xml version="1.0" encoding="UTF-8"?>
 <payment_response>
   <transaction_id>119643250547501c79d8295</transaction_id>
   <usage>40208 concert tickets</usage>
```

```
    <remote_ip>245.253.2.12</remote_ip>
    <amount>100</amount>
    <currency>USD</currency>
    <token>ee946db8-d7db-4bb7-b608-b65b153e127d</token>
    <card_holder>Travis Pastrana</card_holder>
    <cvv>834</cvv>
    <expiration_month>12</expiration_month>
    <expiration_year>2026</expiration_year>
    <customer_email>travis@example.com</customer_email>
    <customer_phone>+1987987987987</customer_phone>
    <account_owner>
      <first_name>Travis</first_name>
      <middle_name>Joe</middle_name>
      <last_name>Pastrana</last_name>
      <full_name_match>Partial match</full_name_match>
      <first_name_match>Match</first_name_match>
      <middle_name_match>No match</middle_name_match>
      <last_name_match>Match</last_name_match>
      <type>personal</type>
    </account_owner>
  </payment_response>
```

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| **account_owner** | | |
| first_name | string | Account owner's first name |
| middle_name | string | Account owner's middle name |
| last_name | string | Account owner's last name |
| full_name_match | string | Full name account match decision |
| last_name_match | string | Last name account match decision |
| middle_name_match | string | Middle name account match decision |
| first_name_match | string | First name account match decision |
| type | string | Account owner type |

# Smart Routing

## Introduction

The Smart Routing API is a higher-level abstraction that allows for simpler and more efficient gateway Processing API integration. It does not require the terminal token as part of the URL when creating transactions by providing an automatic routing capability. This by itself minimizes the need for complex customer-level manual routing to terminals set up on the gateway platform configuration layer.

The Smart Routing API requires the same merchant credentials and performs analogous HTTP BASIC authentication as described under the Authentication section, thus should not add any overhead for managing additional credentials.

## How Smart Routing works

The Smart Routing algorithm analyzes the submitted API request and its data and matches with various parameters and enabled settings to be able to route the transaction to the right terminal under the hood.

#### DESCRIPTION OF THE ALGORITHM

First it's validating that the merchant is allowed to use the Smart Routing capability. After that, an initial selection of terminals and MIDs based on the merchant and the requested currency is made. Once the selection is done, additional filtrations are performed in the following order:

- Based on the transaction type and the MID as well as the terminal-supported transaction types
- Based on the supported features of the terminal
- Based on the supported features of the MID
- Based on the required params of the terminal
- Based on whether the same MID as the reference transaction should be used
- Based on the card brand, only if the card number is present, and if the card brand is supported by the MID

Additionally:

- If exactly one matching gateway configuration is found after each filtration, the only suitable terminal will be used to process the transaction and no further filtrations on the gateway configuration will be executed.
- If, after all filtrations were applied, more than one matching gateway configuration exists, the first terminal from the list will be considered suitable and it will be used to process the transaction.
- If no matching gateway configuration is found (resulting in no suitable terminals to be routed to), a descriptive error will be returned as described in Errors section and the transaction will not be processed successfully.

## How to use Smart Routing

In order to effectively use the Smart Routing API functionality, first it should be enabled through the merchant configuration, thus please open a ticket to Tech Support in case this functionality needs to be tested and/or used. The Smart Routing API is designed to be 100% backward-compatible and the payload sent to the Smart Routing API endpoint should be the same as with the current Processing API endpoints described under the Transactions section.

There are 2 notable differences:

- The new Smart Routing endpoint URL is expectedly different.
- The `Accept` header signifying accepted content types should be submitted, with values of either `Accept: text\xml` or `Accept: application\xml`.

> Our Smart Routing API employs rate and burst limit mechanisms to protect from abuse and ensure smooth, uninterrupted operation at all times. if you have special capacity/TPS requirements, please open a ticket to Tech Support.

# Installments

Installments-related data is supplied as optional data to the standard API request. If a valid installment is supplied, then the transaction will be processed. Otherwise, the transaction will **NOT** be processed successfully and will return proper error code and messages.

Installments are supported for Authorize, Authorize3D, Capture, Sale, Sale3D, Refund.

> Reference transactions support installments only by using the referenced transaction installment. They only inherit the chosen plan without any option to modify it.

Installments are specific per acquirer and card brand. If you want to use installments, please contact [tech-support@e-comprocessing.com](mailto:tech-support@e-comprocessing.com).

---

# Fetch Applicable Installments

Fetch the applicable installments for a given amount, currency and card brand.

`POST /v1/installments`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/installments \
  -X POST \
  -H "Content-Type: application/json" \
  -d '
 {
   "amount": "8000",
   "currency": "GBP",
   "card_number": "4005522731304879"
 }'
```

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| amount | required | integer > 0 | Amount of transaction in minor currency unit, see Currency and Amount Handling for details |
| currency | required | string(3) | Currency code in ISO 4217 |
| card_number | required | 13 to 16 digits | Complete cc number of customer |

`required* = conditionally required`

Successful Response


```
 {
   "plans": [
     {
       "id": "ZXlKaGJHY2lPaUpJVXpJMU5pSjkuZXlKcmJHRnpjeUk2SWtkbGJtVnphWE5KYm5OMFlXeHNiV1Z1ZEhNNk9sWnBjMkU2T2tsdWMzUmhiR3h0Wlc1MFVHeGhiaUlzSW1sa0lqbzNmUS5WU3pybXBXNldSNlRQODNSOGVGWjR3TzNHcVYwVWJicEN4Z3Rqc1p2YzRZ",
       "frequency": "MONTHLY",
       "installments": 12,
       "cost_info": {
         "annualPercentageRate": 0,
         "feeInfo": [
           {
             "type": "CONSUMER",
             "ratePercentage": 400,
             "flatFee": 0,
             "monthlyRatePercentage": 33.33
           },
           {
             "type": "MERCHANT_FUNDING",
             "ratePercentage": 0,
             "flatFee": 0
           },
           {
             "type": "MERCHANT_SERVICE",
             "ratePercentage": 20
           }
         ],
         "totalPlanCost": 84240,
         "totalFees": 3240,
         "totalUpfrontFees": 0,
         "totalRecurringFees": 3240,
         "firstInstallment": {
           "upfrontFee": 0,
           "installmentFee": 270,
           "amount": 6750,
           "totalAmount": 7020,
           "totalAmountExcludingUpfrontFee": 7020
         },
         "lastInstallment": {
           "installmentFee": 270,
           "amount": 6750,
           "totalAmount": 7020
         },
         "currency": "GBP"
       },
       "data": {
         "name": "UK12M0400",
         "type": "ISSUER_DEFAULT",
         "termsAndConditions": [
           {
             "url": "https://www.hsbc.co.uk/help/card-support/instalments",
             "version": 3,
             "text": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla viverra sapien sit amet risus placerat, nec molestie velit cursus. Aliquam bibendum sem eu quam finibus, in vestibulum odio lobortis. Ut at quam eget enim dapibus elementum vitae id enim. Ut nec
             "languageCode": "eng"
           }
         ],
         "promotionInfo": {
           "promotionCode": "P2",
           "promotionID": "PR12"
         },
         "fundedBy": [
           "CONSUMER"
         ]
       },
       "created_at": "2024-10-04T10:55:38.121Z",
       "updated_at": "2024-10-04T10:55:38.121Z",
       "grouping_id": "ZXlKaGJHY2lPaUpJVXpJMU5pSjkuZXlKcmJHRnpjeUk2SWtkbGJtVnphWE5KYm5OMFlXeHNiV1Z1ZEhNNk9sWnBjMkU2T2tsdWMzUmhiR3h0Wlc1MFVHeGhia2R5YjNWd2FXNW5JaXdpYVdRaU9qVXpmUS5uTnd4eG9uZmVGUVlxOHA1UWFCUFFobnpHRGNjT0w5ekg1LUo3blZtd25B"
     }
   ],
   "applicable_terminal_tokens": [
     "TERMINAL-TOKEN1",
     "TERMINAL-TOKEN2"
   ]
 }
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| **plans** | | |
| id | string | The unique identifier of the installment that will be used for the processing |
| frequency | string | The frequency of the installment. Can be `WEEKLY`, `BIWEEKLY`, `MONTHLY`, and `BIMONTHLY`. |

| | | |
|---|---|---|
| installments | integer | The count of installments. |
| cost_info | object | Contains all cost related information like rates, fees, total cost, currency, first and last installments. |
| data | object | Contains various additional data, including the terms and conditions of the installment. |
| created_at | yyyy-mm-dd hh:mm:ss | Represents the creation date time of the installment. |
| updated_at | yyyy-mm-dd hh:mm:ss | Represents the last update date time of the installment. |
| grouping_id | string | Identifier of the installment group that the current installment is part of. |
| applicable_terminal_tokens | array | List of terminal tokens that can be used to process transaction with any of the returned installment plans |

## Show Installment Details

Retrieves details about a specific installment based on its ID.

`GET /v1/installments/<INSTALLMENT_ID>`

Request

```
curl https://username:53ef627e5f9ac4766a8b872a2e0a8548370f0cd2@staging.gate.e-comprocessing.net/v1/installments/<INSTALLMENT_ID> \
  -X POST \
```

Successful Response


```
{
  "installment_plan": {
    "id": "ZXlKaGJHY2lPaUpJVXpJMU5pSjkuZXlKcmJHRnpjeUk2SWtkbGJtVnphWE5KYm5OMFlXeHNiV1Z1ZEhNNk9sWnBjMkU2T2tsdWMzUmhiR3h0Wlc1MFVHeGhiaUlzSW1sa0lqbzNmUS5WU3pybXBXNldSNlRQODNSOGVGWjR3TzNHcVYwVWJicEN4Z3Rqc1p2YzRZ",
    "frequency": "MONTHLY",
    "installments": 12,
    "cost_info": {
      "annualPercentageRate": 0,
      "feeInfo": [
        {
          "type": "CONSUMER",
          "ratePercentage": 400,
          "flatFee": 0,
          "monthlyRatePercentage": 33.33
        },
        {
          "type": "MERCHANT_FUNDING",
          "ratePercentage": 0,
          "flatFee": 0
        },
        {
          "type": "MERCHANT_SERVICE",
          "ratePercentage": 20
        }
      ],
      "totalPlanCost": 84240,
      "totalFees": 3240,
      "totalUpfrontFees": 0,
      "totalRecurringFees": 3240,
      "firstInstallment": {
        "upfrontFee": 0,
        "installmentFee": 270,
        "amount": 6750,
        "totalAmount": 7020,
        "totalAmountExcludingUpfrontFee": 7020
      },
      "lastInstallment": {
        "installmentFee": 270,
        "amount": 6750,
        "totalAmount": 7020
      },
      "currency": "GBP"
    },
    "data": {
      "name": "UK12M0400",
      "type": "ISSUER_DEFAULT",
      "termsAndConditions": [
        {
          "url": "https://www.hsbc.co.uk/help/card-support/instalments",
          "version": 3,
          "text": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla viverra sapien sit amet risus placerat, nec molestie velit cursus. Aliquam bibendum sem eu quam finibus, in vestibulum odio lobortis. Ut at quam eget enim dapibus elementum vitae id enim. Ut nec co
          "languageCode": "eng"
        }
      ],
      "promotionInfo": {
        "promotionCode": "P2",
        "promotionID": "PR12"
      },
      "fundedBy": [
        "CONSUMER"
      ]
    },
    "created_at": "2024-10-04T10:55:38.121Z",
    "updated_at": "2024-10-04T10:55:38.121Z",
    "grouping_id": "ZXlKaGJHY2lPaUpJVXpJMU5pSjkuZXlKcmJHRnpjeUk2SWtkbGJtVnphWE5KYm5OMFlXeHNiV1Z1ZEhNNk9sWnBjMkU2T2tsdWMzUmhiR3h0Wlc1MFVHeGhia2R5YjNWd2FXNW5JaXdpYVdRaU9qVXpmUS5uTnd4eG9uZmVGUVlxOHA1UWFCUFFobnpHRGNjT0w5ekg1LUo3blZtd25B"
  }
}
```


**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| **installment_plan** | | |
| id | string | The unique identifier of the installment that will be used for the processing |
| frequency | string | The frequency of the installment. Can be `WEEKLY`, `BIWEEKLY`, `MONTHLY`, and `BIMONTHLY`. |
| installments | integer | The count of installments. |
| cost_info | object | Contains all cost related information like rates, fees, total cost, currency, first and last installments. |
| data | object | Contains various additional data, including the terms and conditions of the installment. |
| created_at | yyyy-mm-dd hh:mm:ss | Represents the creation date time of the installment. |
| updated_at | yyyy-mm-dd hh:mm:ss | Represents the last update date time of the installment. |
| grouping_id | string | Identifier of the installment group that the current installment is part of. |

# Billing Transactions API

## General Info

`POST /billing_transactions/v1/graphql`

Billing Transactions API allows to programmatically retrieve information about billing transactions using given filters, pagination, and sort parameters.
It allows to filter billing transactions by transaction unique identifier, billing statement identifiers, and date range.
It supports pagination by page number and allows to set number of entries per page.
It supports sorting by field and direction.

Billing Transactions API is GraphQL-based and is following GraphQL standards/conventions.
It expects GraphQL query as a request body and corresponding `application/graphql` or `application/json` Content-type header to be set.
It has a single endpoint and uses POST method for data retrieval. It allows to retrieve only requested fields and by limiting retrieved fields in response, performance and thus API response time can be improved.
It returns detailed error descriptions in GraphQL-specific error format.

Billing Transactions API has authorization layer, which returns authorization specific error status codes in case of access failure, like: 401/403.

Additionally to given documentation, GraphQL schema can be fetched using a special GraphQL introspection feature.

## Schema and introspection

The Billing Transactions API support a GraphQL's introspection feature, which allows to fetch GraphQL schema.
GraphQL schema is the most comprehensive and native way to describe GraphQL-based APIs.
It contains description of all types, queries and their hierachy/relationships available for given API.

API platform tools, like Postman or Insomnia support automatic GraphQL schema fetching using API introspection feature and GraphQL query autocompletion based on downloaded schema, so it is recommended to use one of those tools or anything similar for analyzing and testing the Billing Transactions API. Please note that introspection feature also requires authorization.

More information about GraphQL schema and introspection can be found on GraphQL's official website:
https://graphql.org/learn/schema/
https://graphql.org/learn/introspection/

## Authorization

API requires authorization with a token issued per merchant.

In order to pass authorization, the merchant must do request with populated `Authorization` header.
Following `Authorization` header value formats are allowed:
- Bearer "billing_api_token"
- Bearer billing_api_token
- Bearer token="billing_api_token"
- Bearer token=billing_api_token
- Token token="billing_api_token"
- Token token=billing_api_token
- Token "billing_api_token"
- Token billing_api_token

`billing_api_token` is a token obtained from tech-support@e-comprocessing.com, which is issued individually per merchant entity.

> You must contact tech-support@e-comprocessing.com in order to obtain billing transactions API token.

#### AUTHORIZATION ERROR RESPONSE STATUS CODES

If the request did not pass authorization, following response status codes might be returned:

| Status Code | Summary |
|---|---|
| 401 | Unauthorized. Either Authorization header is not provided or service could not identify merchant with provided token. |
| 403 | Access Denied. Service identified the merchant, but merchant does not have enough permissions to access target API. |
| 500 | Internal Service Error. Unknown system error. Please contact support. |

If one of the following status codes is returned despite issued token being provided in correct header and in correct format, merchant should contact tech-support@e-comprocessing.com.
If for previously working billing transactions API token, status code 403 started to be returned, then it expired, and new one should be issued.

## Billing transactions retrieval

Filter By Billing Transaction Unique I Ds

Request

```
curl https://staging.api.e-comprocessing.net/billing_transactions/v1/graphql \
  -X POST \
  -H "Accept: application/json" \
  -H "Content-Type: application/graphql" \
  -d '
   query {
     billingTransactions(filter: { uniqueId: ["abc123"] })
     {
       items
       {
         billingStatement
         uniqueId
         transactionType
         billingAmount
         billingCurrency
         transactionDate
         merchantTransactionId
         paymentType
       }
     }
   }'
```

Successful Response

```
{
  "data": {
    "billingTransactions": {
      "items": [
```

```
        {
          "billingStatement": "A123",
          "uniqueId": "abc123",
          "transactionType": "Sale Approved",
          "billingAmount": 24.25,
          "billingCurrency": "EUR",
          "transactionDate": "2023-10-01 18:00:14",
          "merchantTransactionId": "some_id",
          "paymentType": "Sale"
        }
      ]
    }
  }
}
```

Filter By Billing Statement I Ds

Request

```
curl https://staging.api.e-comprocessing.net/billing_transactions/v1/graphql \
  -X POST \
  -H "Accept: application/json" \
  -H "Content-Type: application/graphql" \
  -d '
   query {
     billingTransactions(filter: { billingStatement: ["A123", "B456"] })
     {
       items
       {
         billingStatement
         uniqueId
         transactionType
         billingAmount
         billingCurrency
         transactionDate
         merchantTransactionId
         paymentType
       }
     }
   }'
```

Successful Response

```
{
  "data": {
    "billingTransactions": {
      "items": [
        {
          "billingStatement": "A123",
          "uniqueId": "abc123",
          "transactionType": "Sale Approved",
          "billingAmount": 24.25,
          "billingCurrency": "EUR",
          "transactionDate": "2023-10-01 18:00:14",
          "merchantTransactionId": "some_id_1",
          "paymentType": "Sale"
        },
        {
          "billingStatement": "A123",
          "uniqueId": "def456",
          "transactionType": "Chargeback",
          "billingAmount": 10.0,
          "billingCurrency": "EUR",
          "transactionDate": "2023-10-01 11:50:21",
          "merchantTransactionId": "some_id_2",
          "paymentType": "Sale 3D"
        },
        {
          "billingStatement": "B456",
          "uniqueId": "g7h8i9",
          "transactionType": "Settlement Approved",
          "billingAmount": 9.99,
          "billingCurrency": "USD",
          "transactionDate": "2023-10-03 00:23:55",
          "merchantTransactionId": "some_id_3",
          "paymentType": "Cash"
        }
      ]
    }
  }
}
```

Filter By Transaction Type

Request

```
curl https://staging.api.e-comprocessing.net/billing_transactions/v1/graphql \
  -X POST \
  -H "Accept: application/json" \
  -H "Content-Type: application/graphql" \
  -d '
   query {
     billingTransactions(filter: { transactionType: Settlement Approved })
     {
       items
       {
         billingStatement
         uniqueId
         transactionType
         billingAmount
         billingCurrency
         transactionDate
         merchantTransactionId
         paymentType
       }
     }
   }'
```

Successful Response

```
{
  "data": {
    "billingTransactions": {
      "items": [
        {
          "billingStatement": "B456",
          "uniqueId": "g7h8i9",
          "transactionType": "Settlement Approved",
          "billingAmount": 9.99,
          "billingCurrency": "USD",
          "transactionDate": "2023-10-03 00:23:55",
          "merchantTransactionId": "some_id",
          "paymentType": "Sale 3D"
```

```
        }
      ]
    }
  }
}
```

Use Json Content Type

Request

```
curl https://staging.api.e-comprocessing.net/billing_transactions/v1/graphql \
  -X POST \
  -H "Accept: application/json" \
  -H "Content-Type: application/json" \
  -d '
 {
   "query": "query { billingTransactions(filter: { uniqueId: [\"abc123\"] }) { items { billingStatement uniqueId transactionType } } }"
 }'
```

**REQUEST HEADERS**

| Header | Required | Value |
|---|---|---|
| Content-Type | required | `application/graphql` or `application/json` |
| Accept | required | `application/json` |
| Authorization | required | Token `billing_api_token`. See authorization section |

**REQUEST FILTER PARAMETERS**

Resulting billing transactions collection can be filtered using following filter parameters.
At least one parameter should be provided, but any combination can also work.
If startDate is provided, then endDate should also be provided and vice versa.
Date only format 'yyyy-mm-dd' equals to following date time format: yyyy-mm-dd 00:00:00.

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| startDate | required* | 'yyyy-mm-dd hh:mm:ss' or 'yyyy-mm-dd' | Start of the requested date range. Includes billing transactions, which were created after provided time or at the same moment. Max start and end date difference allowed is 7 days. |
| endDate | required* | 'yyyy-mm-dd hh:mm:ss' or 'yyyy-mm-dd' | End of the requested date range. Includes billing transactions, which were created before provided time. Max start and end date difference allowed is 7 days. |
| uniqueId | optional | list of strings | List of billing transaction unique IDs. Max number of elements is 10000 |
| billingStatement | optional | list of strings | List of billing statement names. Max number of elements allowed is 10. |
| merchantTransactionId | optional | list of strings | List of merchant transaction IDs. Max number of elements allowed is 10000. |
| masterAccountName | optional | list of strings | List of master account names. Max number of elements allowed is 100. |
| transactionType | optional | string(255) | Billing transaction type. See available billing transaction type list below. To be passed with spaces and capital letters if any. |

`required* = conditionally required`

Billing transaction type field list:
- "Authorisation Approved"
- "Authorisation Declined"
- "Settlement Approved"
- "Settlement Declined"
- "Sale Approved"
- "Sale Declined"
- "Refund Approved"
- "Refund Declined"
- "Payout Approved"
- "Payout Declined"
- "CFT Other Approved"
- "CFT Declined"
- "Chargeback"
- "Chargeback Reversal"
- "Chargeback Representment"
- "Retrieval Request"
- "Void"
- "Visa EU Intraregional Fraud Chargeback"
- "MasterCard EU Region Chargeback"
- "Second Chargeback"
- "CFT Visa Approved"
- "CFT MasterCard Approved"
- "Visa RDR"
- "RDR Reversal"
- "Gateway Transaction Fee"
- "Risk Management Transaction Fee"
- "Fraud Engine Transaction Fee"

**REQUEST PAGING AND ORDER PARAMETERS**

Resulting billing transactions collection can be divided into pages and ordered using following parameters.
Single page can be downloaded per request along with information about other pages.
In case 'asc' sort direction is used on field, which can contain `null` value, `null` entries will appear first.
In case 'desc' sort direction is used on field, which can contain `null` value, `null` entries will appear last.

**Request Parameters**

| Parameter | Required | Format | Description |
|---|---|---|---|
| page | optional | integer | Requested page number. Pages enumeration starts from 1. Default value: 1 |
| perPage | optional | integer | Max number of entries per page. Expected value is in range from 1 to 100. Default value: 100 |
| orderByDirection | optional | string(enum) | Direction result collection is sorted by. Possible values: asc, desc. Default value: asc |
| orderByField | optional | string(enum) | Field result collection is sorted by. Default value: transactionDate. See available field list below. |

`required* = conditionally required`

Order by field list:
- billingStatement
- transactionDate
- transactionCurrency
- transactionAmount
- exchangeRate

- billingAmount
- transactionFeeAmount
- commissionPercent
- commissionAmount
- interchangeFee
- region
- settlementStatus

Filter By Date Range And Paginate/Sort

Request

```
curl https://staging.api.e-comprocessing.net/billing_transactions/v1/graphql \
  -X POST \
  -H "Accept: application/json" \
  -H "Content-Type: application/graphql" \
  -d '
   query {
     billingTransactions(filter: { startDate: 2023-10-03, endDate: 2023-10-05 }, paging: { page: 2, perPage: 3 }, sort: { byDirection: desc, byField: transactionDate })
     {
       items
       {
         billingStatement
         uniqueId
         transactionType
         billingAmount
         billingCurrency
         transactionDate
       }
       paging
       {
         page
         perPage
         pagesCount
         totalCount
       }
     }
   }'
```

Successful Response

```
{
  "data": {
    "billingTransactions": {
      "items": [
        {
          "billingStatement": "A123",
          "uniqueId": "g7h8i9",
          "transactionType": "Settlement Approved",
          "billingAmount": 9.99,
          "billingCurrency": "USD",
          "transactionDate": "2023-10-04 23:00:00"
        },
        {
          "billingStatement": "A456",
          "uniqueId": "def456",
          "transactionType": "Chargeback",
          "billingAmount": 10.0,
          "billingCurrency": "EUR",
          "transactionDate": "2023-10-04 20:40:11"
        },
        {
          "billingStatement": "A123",
          "uniqueId": "abc123",
          "transactionType": "Sale Approved",
          "billingAmount": 24.25,
          "billingCurrency": "EUR",
          "transactionDate": "2023-10-03 10:30:00"
        }
      ],
      "paging": {
        "page": 2,
        "perPage": 3,
        "pagesCount": 10,
        "totalCount": 28
      }
    }
  }
}
```

**RESPONSE FIELDS**

If request is successful, response should contain billing transactions list with requested fields and status code 200.
Paging information also needs to be explicitly requested.

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| id | integer | Transaction identifier. |
| uniqueId | string | Transaction unique identifier. |
| billingStatement | string | Name of the billing statement the transaction belongs to. |
| arn | string* | Acquirer reference number. |
| transactionType | string | Transaction type. See list of possible values in `transactionType` filter description |
| transactionDate | yyyy-mm-dd hh:mm:ss* | Date and time when transaction was executed in UTC. |
| transactionCurrency | string(3) | The processing currency code of the transaction in ISO 4217. |
| transactionAmount | float(precise) | The transaction amount (in processing currency). |
| exchangeRate | float(precise) | The applied exchange rate to convert from the processing to the billing currency of the transaction. |
| billingCurrency | string(3) | Billing currency code in ISO 4217. |
| billingAmount | float(precise)* | The calculated billing amount of the transaction. |
| transactionFeeCurrency | string(3) | The currency code of the applied transaction fee according in ISO 4217. |
| transactionFeeAmount | float(precise)* | The applied transaction fee amount. |
| transactionFeeChargedOnBillingStatement | string* | Name of the billing statement the transaction fee for this transaction is deducted from. |
| commissionPercent | float(precise)* | The applied commission (discount) percent. Empty for all types except Sale Approved and Settlement Approved. |
| commissionAmount | float(precise)* | The applied commission (discount) amount in billing currency. Empty for all types except Sale Approved and Settlement Approved. |

| | | |
|---|---|---|
| interchangeFee | float(precise)* | The interchange fee amount (in interchange currency). |
| interchangeCurrency | string(3)* | The interchange currency code of the transaction in ISO 4217. |
| isInterchangeplusplus | boolean | Identifies if the transaction is Interchange++. |
| interchangeplusplusChargedOnBillingStatement | string* | The name of billing statement, Interchange++ is charged on. |
| schemeFee | float(precise)* | The scheme fee amount (in scheme fee currency). |
| schemeFeeCurrency | string(3)* | The scheme fee currency of the transaction. |
| standardDebitCardRate | float(precise)* | The standard debit card rate of the transaction. |
| gstAmount | float | The GST tax calculated amount of the transaction(based on transaction fee and commission). |
| gstRate | float | The applied GST over (transaction fee and discount) percent. |
| vatAmount | float | The VAT tax calculated amount of the transaction(based on transaction fee and commission). |
| vatRate | float | The applied VAT over (transaction fee and discount) percent. |
| terminalName | string* | Associated terminal name. |
| region | string* | Associated region. |
| settlementBillingStatements | string* | Associated billing statement name(s), which partially or fully settled the transaction. Can be multiple statement names separated with comma. |
| settlementDates | yyyy-mm-dd* | Dates when transaction was partially or fully settled. Can be multiple dates separated with comma. |
| settlementStatus | string(enum)* | Transaction settlement status. Possible values: Delayed, Balanced or not defined. |
| merchantId | integer | Merchant unique identifier. |
| merchantName | string | Merchant name. |
| merchantTransactionId | string* | Transaction identifier provided by merchant. |
| masterAccountName | string | Master account name. |
| valueDate | yyyy-mm-dd* | Indicates when the corresponding billing statement was set to Paid status. Will be empty for billing statements with Status Pending. |
| documentId | string* | Consumer personal identification. See Document ID Parameter. |
| referenceId | string* | Reference (original) transaction unique identifier. |
| authCode | string* | Authorization code generated by the card network when an authorisation has occurred. It is used to identify authorization and it consists of 6 alphanumeric chars. |
| paymentType | string* | Reference (original) transaction type. |
| cardBrand | string* | Card brand. Examples: Visa, MasterCard, Maestro |
| cardNumber | string* | Masked card number. Example: 420000...0000 |
| cardHolder | string* | Full name of customer as printed on card (first name and last name at least). Example: John Doe |
| cardType | string* | Card type. Possible values: Debit, Credit, Charge Card |
| cardSubtype | string* | Card subtype. Examples: Standard, Gold, Gift |

`*` means that field can potentially contain `null` value.
`float(precise)` is float type, which maintains certain precision, which is typical for exchange rates, fee amounts, percentages and etc. It is achieved by either rounding up original number to specific precision or by adding trailing zeros for consistency. It is useful for displaying to user. Please note that trailing zeroes might be removed automatically by API platform tools, like Postman.
Examples:
- 1.000000 or 0.023000 for exchangeRate
- 0.0262 or 0.0000 for schemeFee
- 15.29 or 1.10 for transactionAmount

By limiting retrieved fields in response, performance and thus API response time can be improved.

Fee and amount fields contain precise value in the corresponding currency and no exponent (scaling) needs to be applied.

Select All Fields

Request

```
curl https://staging.api.e-comprocessing.net/billing_transactions/v1/graphql \
  -X POST \
  -H "Accept: application/json" \
  -H "Content-Type: application/graphql" \
  -d '
   query {
     billingTransactions(filter: { uniqueId: "["abc123"]" })
     {
       items
       {
         id
         billingStatement
         uniqueId
         arn
         transactionDate
         transactionType
         transactionCurrency
         transactionAmount
         exchangeRate
         billingCurrency
         billingAmount
         transactionFeeCurrency
         transactionFeeAmount
         transactionFeeChargedOnBillingStatement
         commissionPercent
         commissionAmount
         interchangeFee
         interchangeCurrency
         isInterchangeplusplus
         interchangeplusplusChargedOnBillingStatement
         schemeFee
         schemeFeeCurrency
         standardDebitCardRate
         gstAmount
         gstRate
         vatAmount
         vatRate
         terminalName
         region
         settlementBillingStatements
```

```
        settlementDates
        settlementStatus
        merchantId
        merchantName
        merchantTransactionId
        masterAccountName
        valueDate
        documentId
        referenceId
        authCode
        paymentType
        cardBrand
        cardNumber
        cardHolder
        cardType
        cardSubtype
      }
    }
  }`
```

#### PAGING RESPONSE FIELDS

**Successful Response Parameters**

| Parameter | Type | Description |
|---|---|---|
| page | integer | Current page number. Pages enumeration starts from 1. |
| perPage | integer | Max number of entries per each page. |
| pagesCount | integer | Total number of pages associated with particular request filter set. |
| totalCount | integer | Total number of entries associated with particular request filter set. |

It is recommended to fetch `pagesCount` and/or `totalCount` with the first page of data and request subsequent pages w/o using these fields. This will significantly improve subsequent requests performance.

#### RESPONSE ERROR CODES

If request fails, following error codes and messages might be returned:

| Error code | Error message | Response status code | Summary |
|---|---|---|---|
| GRAPHQL_PARSE_FAILED | invalid request syntax/schema | 422 | Invalid GraphQL syntax/schema in request from the billing transactions API client. |
| GRAPHQL_VALIDATION_FAILED | invalid request query/types | 422 | Invalid GraphQL query/types in request from the billing transactions API client |
| INVALID_REQUEST_PARAMETERS | invalid request parameters | 200 | Invalid input parameters values in request from the billing transactions API client, e.g. per page entries number exceeding max allowed value. |
| REQUEST_TIMED_OUT | request timed out, retry recommended | 200 | Request took too long to execute and service interrupted it. |
| INTERNAL_SERVICE_ERROR | internal service error | 200 | Unknown system error. Please contact support. |

GraphQL error(s) returned by service has format described in following GraphQL specification: https://spec.graphql.org/draft/#sec-Errors.
Returned error might be additionally supplied with `debug` field, which contains detailed information about the issue.
Response status code 200 might be returned despite error occurred according to GraphQL conventions.

Error Response

```
{
  "data": {
    "billingTransactions": null
  },
  "errors": [
    {
      "message": "invalid request parameters",
      "path": [
        "billingTransactions"
      ],
      "extensions": {
        "code": "INVALID_REQUEST_PARAMETERS",
        "debug": "validating request filter parameters: startDate and endDate fields should be defined together",
        "domain": "billing-transactions-api"
      }
    }
  ]
}
```

Error Response

```
{
  "data": null,
  "errors": [
    {
      "message": "invalid request query/types",
      "locations": [
        {
          "line": 2,
          "column": 59
        }
      ],
      "extensions": {
        "code": "GRAPHQL_VALIDATION_FAILED",
        "debug": "string cannot represent a non string value: 123",
        "domain": "billing-transactions-api"
      }
    },
    {
      "message": "invalid request query/types",
      "locations": [
        {
          "line": 2,
          "column": 86
        }
      ],
      "extensions": {
        "code": "GRAPHQL_VALIDATION_FAILED",
        "debug": "int cannot represent non-integer value: \"abc\"",
        "domain": "billing-transactions-api"
      }
    }
  ]
}
```

Error Response

```
{
  "data": {
    "billingTransactions": null
```

```
  },
  "errors": [
    {
      "message": "internal service error",
      "path": [
        "billingTransactions"
      ],
      "extensions": {
        "code": "INTERNAL_SERVICE_ERROR",
        "debug": "could not read Merchant ID value from request context",
        "domain": "billing-transactions-api"
      }
    }
  ]
}
```