

Integration Manual e.Redde

v1.16



SUMMARY

About.....	5
Glossary.....	5
Libraries.....	6
SDKs.....	7
Modules	7
First Steps.....	7
Authentication and authorization	7
Endpoints	7
Authorization Flow.....	8
Documentation	14
Technical Prerequisites	15
Accreditation.....	15
Digital Rede Certificate	15
Approval and SSL certificate	15
Affiliation and Integration Key.....	16
HTTP Methods.....	16
Return Codes.....	17
Formatting	18
Transactions.....	19
Capture.....	19
Airlines	23
3D Secure 2.0	26
3DS 2.0 - Rede MPI.....	27
Callback.....	36
3DS 2.0 - External MPI	37
Data Only.....	44
Data Only – Rede MPI	45
Data Only – External MPI	51
Table of ECIs.....	54
Zero Dollar.....	56
Cancellation	59
Notification URL	63
Transaction query	64

Cancellation Query.....	76
softDescriptor	80
Dynamic MCC.....	81
Subacquirers and Marketplaces	85
Public.....	86
Transactional message.....	86
Digital wallets.....	91
Staged Digital Wallet Operators (SDWO)	100
Pix.....	112
Definition of Pix.....	112
Pix key registration.....	112
QR Code Pix request	112
Status notification by webhook	115
Query transaction Pix.....	117
Card Brands Tokenization (Capture).....	127
Rede Tokenization of Card Brand	133
First steps	134
Token query	137
Cryptogram	138
Callback from the webhook	140
Integration returns - Token Requestor	141
Token Management.....	144
Sandbox Tutorial	145
Simulate webhook configuration.....	148
Simulate request status	149
Simulate token management query errors	150
Recurrence and Card-on-File	151
Categorizing of card-on-file transactions.....	155
Returns.....	157
Brand Returns	158
Other Returns	167
Card Center Returns.....	170
Integration returns.....	171
3DS Returns.....	178
Cancellation returns.....	180

Brand Fees.....	181
Transaction Reattempts.....	181
Brand Rules	182
Brand Fees.....	185
No use of Zero Dollar	187
Brand rules	187
Brand fees	187
Use of Zero Dollar	188
Brand rules	188
Brand fees	188
Preauthorization	188
Brand rules	188
Brand fees	188
Authenticated and unauthenticated transactions	189
Non-tokenized transactions.....	189
Brand rules	189
Brand fees	190
Staged Digital Wallets	190
Cancelation	190
Sandbox Tutorial	190
Starting Point	190
Cards	190
Simulate Errors.....	192
Simulate transaction with retroactive date	194
Simulate cancellation callback.....	194
Simulate Zero Dollar transaction	195
Simulate 3DS 2.0 transaction - MPI Rede	195
Simulate transactions with brandTid	196
Simulate Returns.....	197
Simulate MAC.....	197
Simulate reattempt returns	198
Security Advice.....	198
Through the Payment Page:	198
Through compromised tokens:.....	199
Appendix 1 - MCCs allowed in Bill Payment operations via Staged Digital Wallets.....	200

e.Redex Developer Integration Manual

The purpose of this manual is to help the developer to integrate their application with e.Redex, listing the functionalities and methods with examples of messages to be sent and received.

About

e.Redex is a practical and secure online payment solution to make sales over the internet with complete peace of mind.

With different types of integration, the solution captures, and processes financial transactions directly through the Rede, that is, without the need for an intermediary, offering payments with credit and debit cards from the main brands in the market. Credit cards are available under Mastercard, Visa, Hiper, Elo, Diners, Sorocred, American Express, Hipercard, JCB, Banescard, Cabal, Mais, Credz; and Mastercard, Visa and Elo debit cards.

In addition, e.Redex offers a series of functionalities to add even more value to its customers' businesses, with a greater focus on sales conversion and greater management control.

The purpose of this manual is to help the developer to integrate their application with e.Redex, listing the functionalities and methods with examples of messages to be sent and received.

The e.Redex solution was developed thinking about the ease for the establishment that wants to use the API without the need to install new systems. The main advantages of using an API are: interoperability between distinct and physically distant applications, portability between different platforms, easy integration, cost reduction for data transport and universal format.

Glossary

To facilitate understanding, we created a glossary with the most used terms related to e-commerce, acquiring and cards.

- **Authorization:** process that raises awareness of the cardholder's credit limit with the issuing bank, generally used for prevention analysis, credit limit analysis and validation of card data used. Authorization does not generate a charge on the buyer's invoice.
- **Capture:** process that confirms an authorized transaction. After capture, the amount is debited from the cardholder's credit, generating the charge on the cardholder's invoice.
- **Cancellation:** the process that returns the amount authorized or captured on the card to the buyer.

- **PV (affiliation number):** identifier code generated by Rede for affiliated establishments. The POS (point of sale) is unique for each establishment.
- **Integration key:** security code generated by Rede used to guarantee the integrity of the transaction. It is part, along with the POS, of the API's authentication credentials.
- **Issuer:** is the financial institution that issues the credit or debit card.
- **Cardholder:** is the owner of the card, the buyer of the product.
- **Establishment:** is the entity responsible for the e-commerce (shop or virtual service).
- **TID:** is the unique identifier of a transaction composed of up to 20 characters generated by Rede. This identifier is not repeated.
- **MPI:** the system responsible for authenticating credit and debit transactions with the issuer.
- **SecureCode:** is MasterCard's secure purchase system that certifies the cardholder with the issuer, validating data that only they and the bank have. It follows the universal 3D Secure standard.
- **Verified by Visa (VBV):** Visa's secure purchase system that certifies the cardholder with the issuer, validating data that only the cardholder and the bank have. It follows the universal 3D Secure standard.
- **Tokenization:** A process by which the card number is replaced with a value called a static token that will be used in all payment transactions.
- **Lifecycle:** Once a card token is generated, it will go through different stages throughout its existence. The Issuer may also request the replacement of the registration data of the physical card linked to that token, for example, due to expiration, fraud, plastic damage, etc.
- **Cryptogram:** is a unique encrypted value that is dynamically generated by the Brand with each transaction, along with the token data.
- **Token expiration date:** it is generated and maintained in the Brand, it is approved during token processing. The token expiration date is generally not the same as the Card expiration date and may be before or after the stated expiration date.
- **TokenizationId:** Unique Rede identifier for the card number tokenization request made by the cardholder.
- **ProvisionedTokenId:** A unique ID associated with a token. The ID is created after the initial provision of a token by the Brand.
- **Postback:** Synchronous API return, for example, in the 3DS authentication flow.
- **Callback:** Asynchronous return from the API, for example, in the cancellation API.

SDKs

Check out the SDKs available for e.Redex.



Modules

Check out the Modules available for e.Redex.



First Steps

Authentication and authorization

The e.Redex APIs use Basic Auth, which is an industry standard for user and application authorization and authentication. This protocol is based on simplifying client development for authorization flows for web, desktop apps, mobile phones, etc.

Endpoints

Endpoints are the URLs that will be used to call a particular service. They may vary depending on the HTTP environment and method.

The composition is carried out as follows:

- Base URL

- API Version
- Service

Environment	URL
Sandbox	https://sandbox-erede.useredecloud.com.br/v1/transactions
Production	https://api.userede.com.br/erede/v1/transactions

For each type of service, a complement will be added to the base URL so that the request can be made afterward. In the course of this documentation, we will list them all.

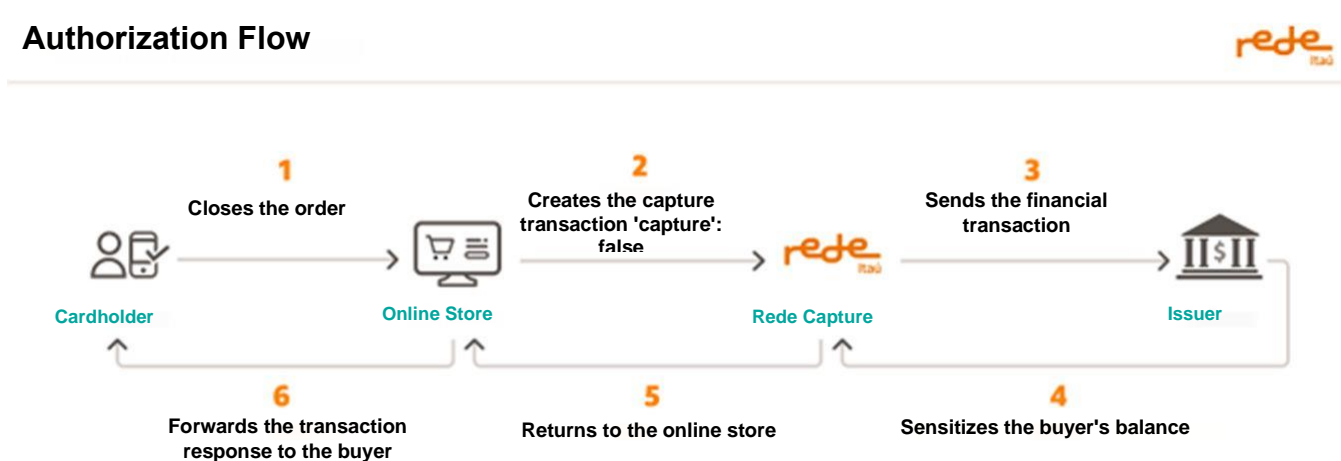
Authorization Flow

Authorization is the first step in carrying out a transaction. The value of the transaction sensitizes the cardholder's limit but does not generate a charge on the invoice until confirmation (capture) occurs.

If the authorization is not captured within the maximum period according to the branch of the establishment, it is automatically cancelled.

For installment transactions, inform the minimum number of "2" and maximum of "12" installments.

The diagram below shows the flow of the permission transaction without automatic capture:



See below in "Request body" with the diversity of possible options.

See below under "Responses" with their respective returns.

For further explanation of each option, see the Documentation further up this page.

POST /v1/transactions

Parameters

Authorization Service Access Token: Basic {{hash_pv_token}}
***required**
Example : Basic
 string NzM4NTQ5Njc6NjA2OWEwMjZjZjQ1NDcwNjk5MGE4MDFhYjVmZThlMzY=
 (header)

Request:

```
{
  "capture": false,
  "kind": "credit",
  "reference": "pedido123",
  "amount": 2099,
  "installments": 2,
  "cardholderName": "John Snow",
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "softDescriptor": "string",
  "subscription": false,
  "origin": 1,
  "distributorAffiliation": 0,
  "brandTid": "string",
  "storageCard": "1",
  "transactionCredentials": {
    "credentialId": "01"
  }
}
```

Requisition parameters:

Name	Size	Type	Mandatory	Description
capture		Boolean	No	Define whether a transaction will be captured automatically or later. Failure to

Name	Size	Type	Mandatory	Description
				<p>submit this field will be considered an automatic capture (true).</p> <p>For debit and Zero Dollar transactions, when this field is sent, the parameter must be set as true, indicating automatic capture.</p>
kind		Alphanumeric	No	<p>Type of transaction to be performed.</p> <ul style="list-style-type: none"> For credit transactions, use credit For debit transactions, use debit <p>Failure to submit this field will be considered credit.</p>
reference	Up to 16	Alphanumeric	Yes	Order code generated by the establishment.
amount	Up to 10	Numeric	Yes	<p>Total transaction amount without thousands and decimal separators.</p> <p>Examples:</p> <ul style="list-style-type: none"> R\$10.00 = 1000 R\$0.50 = 50
installments	Up to 2	Numeric	No	<p>Number of installments in which a transaction will be authorized. From 2 to 12</p> <p>Failure to submit this field will be considered in cash.</p>
cardholderName	Up to 30	Alphanumeric	No	<p>Cardholder's name printed on the card.</p> <p>Do not send special characters</p>
cardNumber	Up to 19	Alphanumeric	Yes	Card number.
expirationMonth	Up to 2	Numeric	Yes	Card expiration month. From 1 to 12.
expirationYear	2 or 4	Numeric	Yes	<p>Card expiration year</p> <p>E.g.: 2028 or 28</p>

Name	Size	Type	Mandatory	Description
securityCode	Up to 4	Alphanumeric	No	The card security code is usually located on the back of the card. Sending this parameter guarantees a greater possibility of approval of the transaction.
softDescriptor	Up to 18 *	Alphanumeric	No	Personalized phrase that will be printed on the cardholder's invoice.
subscription		Boolean	No	<p>Informs the issuer if a transaction comes from a recurrence. If the transaction occurs through a recurrence, send true. Otherwise, send false.</p> <p>Failure to submit this field will be considered the value false.</p> <p>Rede does not manage recurrence schedules, it only allows merchants to indicate whether a transaction originated from a recurring plan.</p>
origin	Up to 2	Numeric	No	<p>Identifies a transaction source.</p> <ul style="list-style-type: none"> e.Redre - 1 <p>Failure to submit this field will be considered an e.Redre (1) transaction.</p>
distributorAffiliation	Up to 9	Numeric	No	Distributor's affiliation number (PV).
brandTid	Up to 16	Alphanumeric	No	<p>Identifier that correlates the first transaction from the subsequent ones.</p> <p>For more details see the section Recurrence and Card-on-file</p>
securityAuthentication	-	-	-	securityAuthentication group

Name	Size	Type	Mandatory	Description
sai	Up to 2	Alphanumeric	Mandatory for Visa and ELO brands. Optional on card-on-file transactions	Electronic Transaction Identifier (ECI). For Mastercard branded transactions, this field is not sent. In transactions that are not tokenized (only card-on-file), sending this field is not necessary. For more details about this field, check the topic "using sai".
transactionCredentials	-	-	-	transactionCredentials Group
transactionCredentials/credentialId	Up to 2	Alphanumeric	Yes, if storagecard=1 or storagecard=2 and is a mastercard brand transaction	Indicates the category of transaction with stored credential. See the section " Categorizing Card-on-File Transactions " for more details.

Use of "sai": The parameter must be used whenever the transaction has a specific ECI, which is not linked to 3DS authentication (ex: Wallets and Cloud Token Visa), when authenticated as 3DS it is necessary that the "eci" is informed within the 3D Secure group, it is not necessary to use the "sai" in this case.

Attention: When sending the threeDSecure group in any request, the "sai" field will be ignored and the 3DS stream will be prioritized.

Response

```
{
  "reference": "pedido123",
  "tid": "8345000363484052380",
  "nsu": "663206341",
  "dateTime": "2017-02-28T08:54:00.000-03:00",
  "amount": 2099,
  "installments": 2,
  "cardBin": "544828",
  "last4": "0007",
  "returnCode": "00",
  "returnMessage": "Success."
}
```

```

"brand": {
  "name": "Mastercard.",
  "returnCode": "82",
  "returnMessage": "Policy (Mastercard use only)",
  "merchantAdviceCode": "03",
  "brandTid": "226332",
  "authorizationCode": "186376",
},
"links": [
  {
    "method": "GET",
    "rel": "transaction",
    "href": "https://sandbox-
eredede.useredecloud.com.br/v1/transactions/8345000363484052380"
  },
  {
    "method": "PUT",
    "rel": "capture",
    "href": "https://sandbox-
eredede.useredecloud.com.br/v1/transactions/8345000363484052380"
  },
  {
    "method": "POST",
    "rel": "refund",
    "href": "https://sandbox-
eredede.useredecloud.com.br/v1/transactions/8345000363484052380/refunds"
  }
]
}

```

Response parameters:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.

Name	Size	Type	Description
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh:mm:ss.TZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones. For more details see the section Recurrence and Card-on-file

Technical Prerequisites

The integration mechanism is simple, so knowledge of web programming language, HTTPS requests and JSON file manipulation are required to successfully deploy the solution.

Accreditation

To request the accreditation of e.Redes and carry out the integration to your application, contact Rede's Call Center:

4001 4433 (*capitals and metropolitan regions*)

0800 728 4433 (*other locations*)

When the accreditation is carried out, the person responsible for the establishment will be notified via email with the affiliation number (PV), guidelines for accessing the Rede's portal and their credentials for integration.

Digital Rede Certificate

What is a digital certificate?

The digital certificate is an electronic file that serves as a virtual identity for a company and through it, online transactions can be carried out with guaranteed authenticity. As a market practice to ensure full protection of the information exchanged between your company and Rede, the Digital Rede Certificate is updated annually.

Why should it be updated? To ensure greater safety in your online sales.

How to update the Digital Rede Certificate? In order to update the certificate within your company, we ask that you direct this activity to your technology team or whoever has access to your server and is responsible for your e-commerce application. If your contact with Rede is made through your platform, gateway or module, we ask you to contact them for the update.

It must be performed from the server that is responsible for the communication between your company and Rede and on which the certificate is already installed. Download the digital certificate according to your operating system determined in the boxes marked below on this page.

To install or update the Digital Rede Certificate, use the link below and follow the instructions.

If your e-commerce does not use a digital certificate, this step is not necessary.

Approval and SSL certificate

In order to transact with the e.Redes API, it is necessary for the merchant to have an SSL security certificate installed on the payment page with 2048-bit encryption or higher, to guarantee the confidentiality of the

transferred information and certify the cardholder that they are actually accessing the desired site, avoiding problems with fraud.

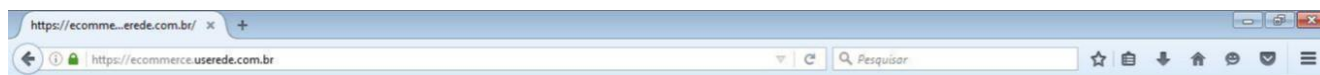
To ensure that the establishments have the SSL certificate installed, Rede carries out the process of automatically approving the store or virtual service of the establishment after the first transaction is carried out.

IMPORTANT: Periodically, the approval process is carried out and Rede reserves the right to suspend the use of the platform until the store or virtual service meets the requested safety standards.

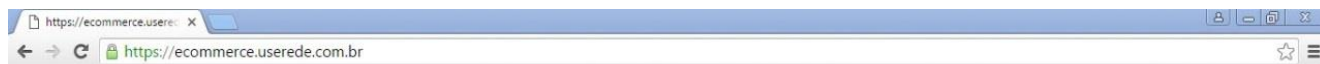
To identify if the page has the SSL certificate, when accessing the site, the URL must be displayed with the "https" protocol, allowing the visualization of the security padlock in browsers.

Examples:

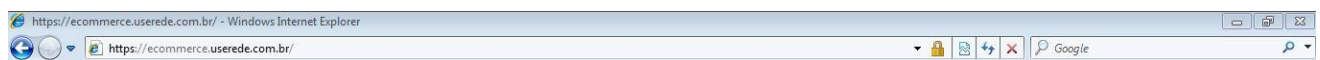
Firefox:



Google Chrome:



Internet Explorer:



If the establishment has been suspended for not being certified, access the Rede portal in the menu *para vender > e-commerce > homologação* and click on "Solicitar homologação" after the SSL certificate has been regularized.

Affiliation and Integration Key

For the merchant to start transacting with e.Redre, it is necessary to configure the API with its credentials: affiliation number (PV) and integration key.

The integration key is a confidential key, generated in the [Rede Portal](#). To generate it, make sure your user has an admin profile (master user). Access the menu: *para vender > e-commerce > chave de integração* and click on "Gerar chave de integração".

In case of loss or forgetfulness of the integration key, a new one must be generated and the API configuration must be changed so that transactions continue to be sent to Rede.

HTTP Methods

HTTP methods for RESTful services will often be used to request transactions.

HTTP VERB	DESCRIPTION
POST	Used when creating resources or sending information that will be processed. For example, creating a transaction.
GET	Used for queries of existing resources. For example, querying transactions.
PUT	Used to update an existing resource. For example, capturing a previously authorized transaction.

The variations will be used according to the requested service: authorization, capture, authorization with automatic capture, query, cancellation and cancellation query.

Return Codes

HTTP return codes are used to indicate the success or failure of an API request. Codes starting with 2xx indicate success, codes starting with 4xx indicate an error due to some incorrect information provided in the request, and codes starting with 5xx indicate an error in the servers.

Success codes

RETURN	DESCRIPTION	METHOD
200	Indicates that the processing was performed correctly and the return will be as expected.	GET
201	Indicates that the resource was created successfully, there must be a header location indicating the url of the new resource.	POST
202	Indicates that the processing will be asynchronous, therefore, in addition to the header location, it must return the content with a status attribute.	POST AND PUT
204	Indicates that the resource was successfully changed or deleted.	PUT

Error Codes

RETURN	DESCRIPTION
400	Poorly formatted request.

RETURN	DESCRIPTION
401	Request requires authentication.
403	Request denied.
404	Resource not found.
405	Method not allowed.
408	Requisition timeout.
413	Request exceeds maximum allowed size.
415	Invalid media type (check request header content-type)
422	Business Exception. Check return code and return message.
429	Request exceeds the maximum number of allowed API calls.

Exception released due to server(s) error

RETURN	DESCRIPTION
500	Server error.

Formatting

Encoding

To use the Rede APIs, it will be necessary to configure the use of UTF-8 encoding in your application.

JSON

JSON (JavaScript Object Notation) is a standard for describing data for exchange between systems. It is simpler and lighter than XML. By default, every API passes JSON, both to receive information (POST and PUT methods) and in return (GET method).

Due to this standardization, for POST and PUT calls it is necessary to inform the HTTP Header content-type: application/json. Otherwise, you will get HTTP error 415: Unsupported Media Type.

Datetime type fields

All attributes of Datetime type, both attributes that are returned in objects and those that are passed as parameters in operations, follow the ISO-8601 standard, represented below:

Date: YYYY-MM-DDThh:mm:ss.sTZD

Example: 2015-11-28T08:54:00.000-03:00

Transactions

Transactions are divided so that the merchant can choose to capture them later or automatically.

In **authorization with subsequent capture**, the transaction value affects the cardholder's card limit, but does not generate a charge on the invoice until there is a confirmation (capture).

In **authorization with automatic capture**, the transaction value is confirmed instantly, without the need to carry out the capture transaction.

Capture

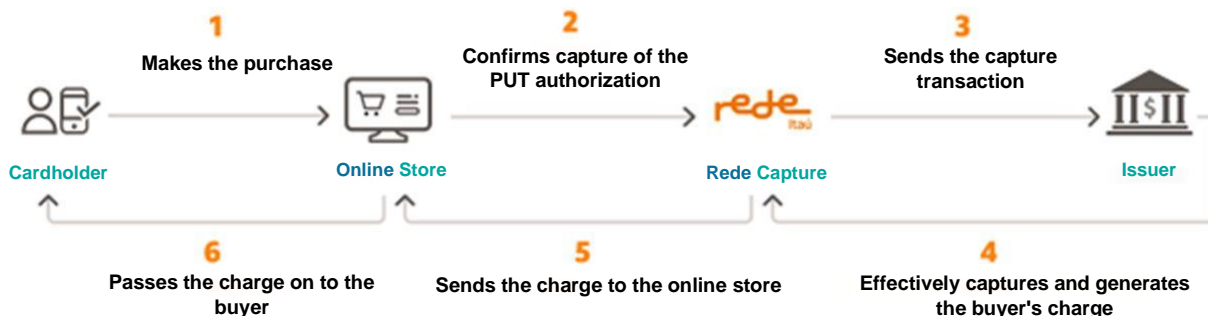
When performing an authorization, confirmation of this transaction (capture) is required. At this point, the charge on the cardholder's invoice is generated.

The authorization must be captured within the maximum period according to the branch of the establishment.

IMPORTANT: Always wait for the transaction response before making a new attempt to capture it.

The diagram below shows the capture transaction flow:

Capture Flow



Confirms the transaction authorization (capture)

When performing an authorization, confirmation of this transaction (capture) is required. At this point, the charge on the cardholder's invoice is generated.

PUT /v1/transactions/{tid}

Parameters

Authorization Service Access Token: Basic {{hash_pv_token}}
***required**
 string
 (header)
Example : Basic NzM4NTQ5Njc6NjA2OWEwMjZjZjQ1NDcwNjk5MGE4MDFhYjVmZThlMzY=

tid *required Unique transaction identifier number. Maximum Size (20).
 string
 (path)
Example : 9274256037511432483

```
{
  "amount": 2099
}
```

Requisition parameters:

Name	Size	Type	Mandatory	Description
amount	Up to 10	Numeric	No	<p>Captures value without thousands and decimal separators. Examples:</p> <ul style="list-style-type: none"> • R\$10.00 = 1000 • R\$0.50 = 50

Response:

```
{
  "reference": "pedido123",
  "tid": "8345000363484052380",
  "nsu": "663206341",
  "dateTime": "2017-02-28T08:54:00.000-03:00",
  "amount": 2099,
  "installments": 2,
  "cardBin": "544828",
  "last4": "0007",
  "returnCode": "00",
  "returnMessage": "Success.",
  "brand": {
    "name": "Mastercard.",
    "returnCode": "82",
    "returnMessage": "Policy (Mastercard use only)",
    "merchantAdviceCode": "03",
    "brandTid": "226332",
    "authorizationCode": "186376",
  },
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://sandbox-
eredede.useredecloud.com.br/v1/transactions/8345000363484052380"
    },
    {
      "method": "PUT",
      "rel": "capture",
      "href": "https://sandbox-
eredede.useredecloud.com.br/v1/transactions/8345000363484052380"
    },
    {
      "method": "POST",
```

```

    "rel": "refund",
    "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/8345000363484052380/refunds"
  }
]
}

```

Response parameters:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh:mm:ss.STZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional

Name	Size	Type	Description
			information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones. For more details see the section Recurrence and Card-on-file

Airlines

Airlines have a different type of transaction, which allows sending the boarding tax amount separately from the air ticket amount. The transaction can be "in cash" or "in installments".

Airline transactions must be of the credit type, with automatic capture (capture = true) and must be sent together with the transaction BODY.

IMPORTANT: Cancellations of IATA type transactions can only be performed after D+1.

POST /v1/transactions

Request:

```
{
  "capture": true,
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "iata": {
    "code": 101010,
    "departureTax": 100
  }
}
```

Requisition parameters:

Name	Size	Type	Mandatory	Description
iata		iata		
iata/code	Up to 9	Numeric	Yes	Airline iata code.
iata/departureTax	Up to 10	Numeric	Yes	Value of the boarding tax without thousands and decimal separators. Examples: <ul style="list-style-type: none"> • R\$10.00 = 1000 • R\$0.50 = 50

Response

```
{
  "reference": "010224113900",
  "tid": "10402402011022100032",
  "nsu": "500022009",
  "dateTime": "2024-02-01T11:39:09-03:00",
  "amount": 60000,
  "installments": 4,
  "cardBin": "544828",
  "last4": "0007",
  "brand": {
    "name": "Mastercard",
    "returnCode": "00",
    "returnMessage": "Success.",
    "authorizationCode": "28943A",
    "brandTid": "MCS2405664610201"
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://api-hom.userede.com.br/eredev1/transactions/10402402011022100032"
    },
    {
      "method": "POST",
      "rel": "refund",

```

```

        "href":
hom.userede.com.br/eredev1/transactions/10402402011022100032/refunds"
    }
]
}

```

Response parameters:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh:mm:ss.TZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional

Name	Size	Type	Description
			information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones. For more details see the section Recurrence and Card-on-file

3D Secure 2.0

3D Secure or 3DS authenticated transactions are transactions that require additional authentication to ensure greater security for the cardholder in online purchases.

3DS authentication is performed by validating data that only the cardholder and the bank have, such as card password, birth date, security code, bank token. **In the case of successful authentication, the issuer assumes the risk of the transaction.**

3D Secure 2.0 is a new authentication standard to provide additional security to transactions and is the first solution capable of authenticating a transaction without customer intervention (no-challenge authentication), as the issuer will have access to more transaction information, not just the value and card data. In cases that require authentication (with challenge) the process is more intuitive and can occur via biometrics, voice/facial recognition or SMS sending, helping to avoid cart abandonment. Who decides whether the transaction should have a challenge or not, is the issuer.

In terms of the market, the Rede currently provides the use of version 2.1 of the 3DS protocol. We are working to make version 2.2 available soon. In short, the 3D Secure 2.0 protocol speeds up authentication, increases security and increases conversion rates, providing shoppers with an extremely smooth quick checkout, especially on mobile, and bringing extra protection to the merchant. See the table of features below:

3DS 2.0 Features	Benefits
Replaces static passwords with 2 strong factors: RBA, OTP, Biometrics or Alternative Channel (Out of Band).	<ul style="list-style-type: none"> • Greater safety • Greater convenience • Less friction

3DS 2.0 Features	Benefits
Support for different payment channels (in-app, IoT, browser, etc).	<ul style="list-style-type: none"> • Best UX • Greater coverage • Improved control for establishments
Purchase support and additional use cases (Card on File provisioning, Digital Wallets, Recurring Payments, Tokenization, etc).	<ul style="list-style-type: none"> • Greater applicability • Greater safety

3DS authentication is mandatory for all debit card transactions. For credit cards, their use is optional.

The MPI (merchant plug-in) is the service that provides the establishment's integration with different issuers, in line with the certifications of the brands for processing 3D Secure (3DS) authentication.

e.Redes offers two ways to use the 3DS service, Rede MPI or External MPI. The use of the MPI will be at the discretion of the establishment.

- Rede MPI: service already embedded in the e.Redes platform, without the need for additional contracting. In this scenario, Rede performs the authentication and authorization flow of the transaction.
- External MPI: service additionally contracted by the customer for integration with e.Redes, without the influence of Rede in the authentication of the transaction. Therefore, in this scenario, Rede performs only the authorization flow.

For 3DS transactions to take place, issuers also need to be prepared to receive buyer authentication information. The main issuers in Brazil already offer this service to their customers.

3DS 2.0 - Rede MPI

Transactions that use the 3DS service with Rede MPI can be of the credit or debit type and must be sent together with the BODY of the authorization transaction.

MPI Rede allows 3DS2.0 authentication in Visa, Mastercard and Elo transactions.

For credit transactions, if the transaction has not been successfully authenticated, the merchant can choose to proceed with the transaction without proper 3DS authentication, and the risk of the transaction passes to the merchant, returning to the "ordinary" transactional cycle.

For debit transactions, the value of this parameter is automatically set due to the authentication requirement.

To enable the service, access the useRede portal, menu: *para vender > e-commerce > 3DS > Contratar*.

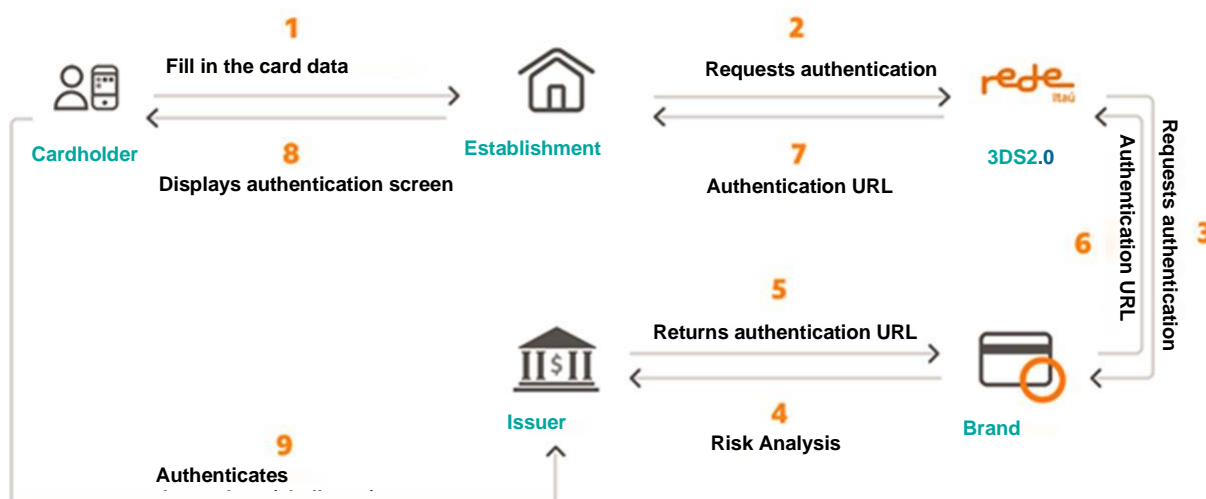
In up to 7 business days, Rede will return informing the status of the service activation request.

The diagrams below show the authenticated (1) and authorized (2) transaction flow using MPI Rede, when the issuer requests a challenge:

Payment Flow 3DS 2.0

Authentication

With challenge

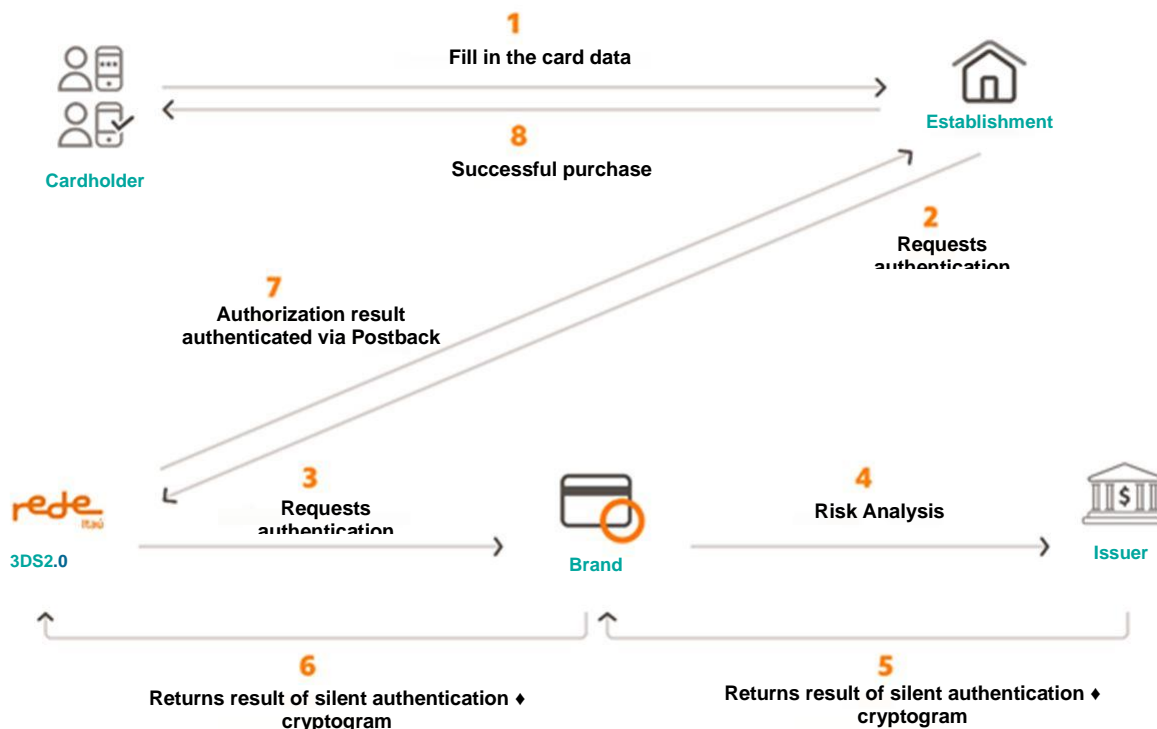


Authentication Flowchart 1 (with challenge)



Authorization Flowchart 2 (with challenge)

The diagram below, on the other hand, illustrates the authenticated and authorized transaction flow when the issuer does **not** request the challenge from the buyer:



Authentication + authorization flowchart 3 (no challenge)

IMPORTANT: To check the transaction status, use the query endpoint. [Click here](#) for more information.

POST /v1/transactions

Request:

```
{
  "capture": false,
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "2223000148400010",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "kind": "credit",
  "threeDSecure": {
    "embedded": true,
    "onFailure": "continue",
    "userAgent": "Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us)
AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405",
    "ipAddress": "192.168.130.20",
    "responseMode": "event",
    "device": {
      "colorDepth": 1,
      "ceviceType3ds": "BROWSER",
      "javaEnabled": false,
      "language": "BR",
      "screenHeight": 500,
      "screenWidth": 500,
      "timeZoneOffset": 3
    }
  },
  "billing": {
    "address": "Rua Pedro Luiz",
    "city": "Guarulhos",
    "postalcode": "07151-385",
    "state": "SP",
    "country": "Brasil",
    "emailAddress": "email@user.com",
    "phonenumber": "(11)91234-5678",
  },
  "urls": [
    {
      "kind": "threeDSecureSuccess",
      "url": "https://scommerce.userede.com.br/LojaTeste/Venda/sucesso"
    },
    {
      "kind": "threeDSecureFailure",
```

```

    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/erro"
  },
  {
    "kind": "threeDSecureCallback",
    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/callback"
  }
]
}

```

Requisition parameters:

Name	Size	Type	Mandatory	Description
threeDSecure		threeDSecure	Yes	
threeDSecure /embedded		Boolean	No	<p>Informs if the MPI service used will be from Rede or a third party.</p> <ul style="list-style-type: none"> true: uses Rede's MPI service false: uses third-party's MPI service <p>Failure to send this field will be considered the use of Rede's MPI.</p>
threeDSecure /onFailure		Alphanumeric	No	<p>Defines how to proceed with the transaction if 3DS authentication is not successful.</p> <ul style="list-style-type: none"> continue: proceeds with the financial transaction even if authentication fails decline: do not proceed with the financial transaction if authentication fails <p>For debit transactions, the value of this parameter is automatically set to decline due to mandatory authentication.</p>

Name	Size	Type	Mandatory	Description
threeDSecure /userAgent	Up to 255	Alphanumeric	Yes	Browser identifier used by the buyer at the time of purchase.
threeDSecure /ipAddress	11	Numeric	Yes	
threeDSecure/responseMode		Alphanumeric	Yes	Field representing the 3DS challenge format: - "event": possibility of displaying the challenge in an iframe
threeDSecure /device				
threeDSecure /device/colorDepth	2	Numeric	Yes	Field representing the estimate of the color palette used for displaying images in bits per pixel. Obtained in the client's browser via the property.
threeDSecure /device/deviceType3ds	20	Alphanumeric	Yes	Field that indicates the type of device on which the authentication takes place.
threeDSecure /device/javaEnabled		Boolean	Yes	Boolean field representing the browser's ability to run Java. The value is the one returned by the navigator.javaEnabled property, true or false.
threeDSecure /device/language	10	Alphanumeric	Yes	Browser language in IETF BCP47 format, containing between 1 and 8 characters.
threeDSecure /device/screenHeight	6	Numeric	Yes	The total height of the customer's screen in pixels. The value is the one returned by the screen.height property.
threeDSecure /device/screenWidth	6	Numeric	Yes	The total width of the client's screen in pixels. The value is the one returned by the screen.width property.
threeDSecure /device/timeZoneOffset	10	Numeric	Yes	Time difference, in hours, between UTC and the cardholder's browser local time.
billing		billing	Yes	Cardholder data

Name	Size	Type	Mandatory	Description
billing /address		Alphanumeric	Yes	Address
billing /city		Alphanumeric	Yes	City
billing /postalcode	8	Numeric	Yes	Postalcode
billing /state		Alphanumeric	Yes	State
billing /country		Alphanumeric	Yes	Country
billing /emailAddress		Alphanumeric	Yes	E-mail
billing /phoneNumber		Numeric	Yes	Phone
urls		urls		
urls/kind		Alphanumeric	Yes	Field that identifies the type of the url. <ul style="list-style-type: none"> • threeDSecureSuccess • threeDSecureFailure • threeDSecureCallback
urls/url	Up to 87	Alphanumeric	Yes	Field to inform the url that the buyer should be redirected after authentication and be notified via callback with the transaction data. Click here for more information.

Attention: 3DS Internal MPI only be used with the following messages:

- [Dynamic MCC](#): Messaging used for operations that transact using multiple merchant category codes;
- [Staged Digital Wallet Operators \(SDWO\)](#)
- [Rede brand tokenization](#)
- [External brand tokenization \(capture\)](#)

These messages can be used together or individually. To see an example of all messages together in a request:

```
{
  "capture": false,
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "2223000148400010",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "kind": "credit",
  "softDescriptor": "LOJADOZE",
  "tokenCryptogram": "ANbuvvxndbK2AAEShHMWgGADFA==",
  "paymentFacilitatorID": 22349202212,
  "independentSalesOrganizationID": 1234567,
  "subMerchant": {
    "mcc": 1234,
    "subMerchantID": 1234567890,
    "city": "Guarulhos",
    "cep": "07151-385",
    "state": "SP",
    "country": "BR",
    "address": "Rua Pedro Luiz",
    "phoneNumber": "(11)91234-5678",
    "cnpj": "71.789.371.0001-42",
    "taxIdNumber": "99999999999"
  },
  "threeDSecure": {
    "embedded": true,
    "onFailure": "continue",
    "userAgent": "Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405",
    "ipAddress": "192.168.130.20",
    "responseMode": "event",
    "device": {
      "colorDepth": 1,
      "deviceType3ds": "BROWSER",
      "javaEnabled": false,
      "language": "BR",
      "screenHeight": 500,
      "screenWidth": 500,
      "timeZoneOffset": 3
    }
  },
  "billing": {
```

```

"address": "Rua Pedro Luiz",
"city": "Guarulhos",
"postalcode": "07151-385",
"state": "SP",
"country": "Brasil",
"emailAddress": "email@user.com",
"phonenumber": "(11)91234-5678",
}
},
"consumerBillPaymentService": {
  "businessApplicationIdentifier": "01"
},
"wallet": {
  "walletId": 1234567890,
  "processingType": "02",
  "senderTaxIdentification": "11122233344"
},
"urls": [
  {
    "kind": "threeDSecureSuccess",
    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/sucesso"
  },
  {
    "kind": "threeDSecureFailure",
    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/erro"
  },
  {
    "kind": "threeDSecureCallback",
    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/callback"
  }
]
}

```

Response:

```

{
  "dateTime": "2019-08-01T10:01:03.000-03:00",
  "threeDSecure": {
    "embedded": true,
    "url":
      "https://scommerce.userede.com.br/adquirencia/mpi/auth?token=7z3nVtKnea4Gs4N%2bpeM2
      j8M%2fund%2bTfwDNVoAZLC7EqKC3gqz0j1DV0%2fpg928%2bh1sABmy1ZPlzyWxTqakpotWa%2f7lq9wsDZ
      6b4Yk8KVXxR1xM00UmxR05V2bZiq%2bF%2fcrscUWE76jS%2fDRokAa1RTEhn18Yw8Q7SAZafPc4YCpdWYgf

```

```
VUxtZP4FDrC2KAqayjzPL85JjYDeujUdCgVhEhWimXLkkH6iJfC2qYPppv94iHm4CY1AyKsxmjkb76KFFOB
4uGocaHSR%2f0gDo0TyTxmxZ%2fTtH45WKq%2b4XmqG9LnDI%3d"
  },
  "returnCode": "220",
  "returnMessage": "Transaction request with authentication received. Redirect URL
sent
```

Response parameters:

Name	Size	Type	Description
dateTime		Datetime	Transaction date in the format YYYY-MM-DDThh:mm:ss.sTZD .
threeDSecure		threeDSecure	
threeDSecure /embedded		Boolean	Informs if the MPI service used will be from Rede or a third party.
threeDSecure /url	Up to 500	Alphanumeric	Authentication url returned by the MPI system.
returnCode	3	Alphanumeric	Return code from 3ds transaction (see 3DS returns table).
returnMessage	Up to 256	Alphanumeric	Return message from 3ds transaction (see 3DS returns table).

Callback

The Callback is an asynchronous return from the API, which will be sent to the endpoint indicated to receive the post method. This “indication” is made in the transaction request itself, in the parameter: "kind":"threeDSecureCallback". The callback will indicate the result of the financial transaction.

Name	Size	Type	Description
reference	Up to 16	Alphanumeric	Order code generated by the establishment.
tid	20	Alphanumeric	Unique transaction identifier number.
nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.

Name	Size	Type	Description
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
expiresAt		Date and Time	Pre-authorization expiration date in format YYYY-MM-DDThh:mm:ssTZD
date		Date	Transaction date in yyyyMMdd format.
time		Time	Transaction time in HH:mm:ss format.
returncode	Up to 4	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.
threeDSecure.returnCode	Up to 4	Alphanumeric	3DS return code (see 3DS returns table).
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand

3DS 2.0 - External MPI

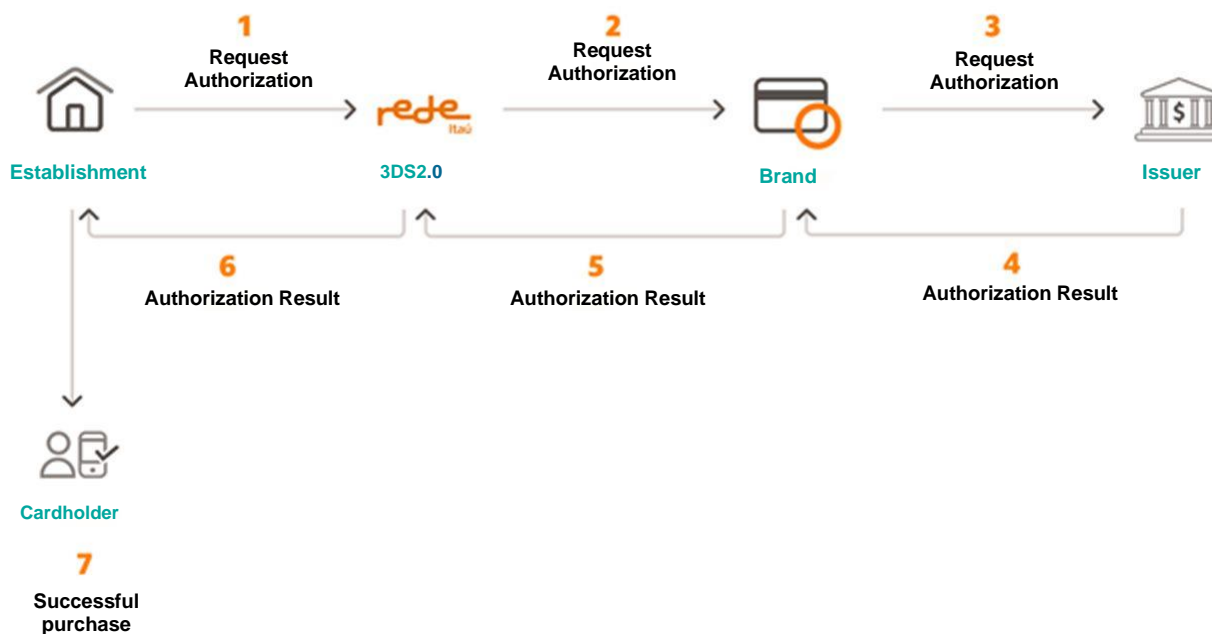
The External MPI is used when the establishment already has a contracted MPI. Therefore, use the *embedded* parameter to signal that the MPI has already been contracted externally to Rede, see the "Request Parameters" table.

For 3DS transactions to be authenticated by the issuer and later authorized using e.Redre, through the external MPI, the MPI service must be certified by the brands and Rede. Currently, certified MPI services are: Lyra, Cardinal and Datacash.

In this external authentication scenario, Rede can receive transactions from all brands, among those already prepared for the product, and thus proceed with the authorization flow.

Transactions that use the 3DS service with customer MPI can be of the credit or debit type and must be sent together with the **BODY** of the authorization transaction.

The diagram below shows the authenticated transaction authorization flow using MPI Client:



Authorization Flowchart 4 (with or without challenge)

POST /v1/transactions

Request:

```
{
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "kind": "credit",
  "threeDSecure": {
    "embedded": false,
    "cavv": "BwABBy1VaQAAAAFwllVpAAAAAAA=",
    "eci": "05",
    "xid": "Nmp3VfDwMlEwZ05pWGN3SGo4TDA=",
    "threeDIndicator": "2",
    "directoryServerTransactionId": "f38e6948-5388-41a6-bca4-b49723c19437"
  }
}
```

Requisition parameters:

Name	Size	Type	Mandatory	Description
threeDSecure		threeDSecure	Yes	
threeDSecure /embedded		Boolean	No	<p>Informs if the MPI service used will be from Rede or a third party.</p> <ul style="list-style-type: none"> • true: uses Rede's MPI service • false: uses third-party's MPI service <p>Failure to send this field will be considered the use of Rede's MPI.</p>
threeDSecure / eci	2	Alphanumeric	Yes	Code returned to the MPI by the Brands that indicates the result of the bearer authentication with the Issuer. It must be sent only for the use of the 3DS authentication service. Debit transactions must be authenticated.
threeDSecure /cavv	Up to 32	Alphanumeric	Yes	Cryptogram code used to authenticate the transaction and sent by the establishment's MPI (may contain special characters). It must be sent only for the use of the 3DS authentication service.
threeDSecure /xid	28	Alphanumeric	*Yes	<p>Authentication transaction ID added by MPI to the establishment (may contain special characters). It must be sent only for the use of the 3DS authentication service.</p> <p>Field used only for * Visa brand.</p>
threeDSecure /threeDIndicator	1	Alphanumeric	Yes	3DS version used in the authentication process by MPI.

Name	Size	Type	Mandatory	Description
threeDSecure /directoryServerTransactionId	36	Alphanumeric	Yes	Authentication transaction ID added by MPI to the establishment (may contain special characters). It must be sent only for the use of the 3DS authentication service.

Response:

```
{
  "reference": "M20240220150251",
  "tid": "10402402201047220605",
  "nsu": "500011185",
  "dateTime": "2024-02-20T15:25:41-03:00",
  "amount": 96000,
  "cardBin": "544828",
  "last4": "0007",
  "brand": {
    "name": "Maestro",
    "returnCode": "00",
    "returnMessage": "Success.",
    "authorizationCode": "380586",
    "brandTid": "MS11675437260220"
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://api-hom.userede.com.br/erede/v1/transactions/10402402201047220605"
    },
    {
      "method": "POST",
      "rel": "refund",
      "href": "https://api-hom.userede.com.br/erede/v1/transactions/10402402201047220605/refunds"
    }
  ]
}
```

Response parameters:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh:mm:ss.TZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones.

Name	Size	Type	Description
			For more details see the section Recurrence and Card-on-file

Attention point:

It is possible to use the Internal 3DS MPI in conjunction with the following messages:

- [Dynamic MCC](#): Specific message for customers who work with more than one MCC;
- [Staged Digital Wallet \(SDWO\)](#)
- [Tokenization Rede](#)
- [Card Brands Tokenization \(capture\)](#)

Messages can be used together or individually.

See an example below of all messages together in the request:

```
{
  "capture": false,
  "kind": "credit",
  "reference": "pedido123",
  "amount": 2000,
  "cardholderName": "John Snow",
  "cardNumber": "2223000148400010",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "softDescriptor": "LOJADOZE",
  "tokenCryptogram": "ANbuvvxndbK2AAEShHMWGgADFA==",
  "paymentFacilitatorID": 22349202212,
  "independentSalesOrganizationID": 1234567,
  "subMerchant": {
    "mcc": 1234,
    "subMerchantID": 1234567890,
    "city": "Guarulhos",
    "cep": "07151-385",
    "state": "SP",
    "country": "BR",
    "address": "Rua Pedro Luiz",
    "phoneNumber": "(11)91234-5678",
    "cnpj": "71.789.371.0001-42",
    "taxIdNumber": "99999999999"
  }
},
```

```

"threeDSecure": {
  "embedded": true,
  "onFailure": "continue",
  "userAgent": "Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us)
AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405",
  "ipAddress": "192.168.130.20",
  "device": {
    "colorDepth": 1,
    "deviceType3ds": "BROWSER",
    "javaEnabled": false,
    "language": "BR",
    "screenHeight": 500,
    "screenWidth": 500,
    "timeZoneOffset": 3
  },
  "billing": {
    "address": "Rua Pedro Luiz",
    "city": "Guarulhos",
    "postalCode": "07151-385",
    "state": "SP",
    "country": "Brasil",
    "emailAddress": "email@user.com",
    "phoneNumber": "(11)91234-5678"
  }
},
"consumerBillPaymentService": {
  "businessApplicationIdentifier": "01"
},
"wallet": {
  "walletId": 1234567890,
  "processingType": "02",
  "senderTaxIdentification": "11122233344"
},
"urls": [
  {
    "kind": "threeDSecureSuccess",
    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/sucesso"
  },
  {
    "kind": "threeDSecureFailure",
    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/erro"
  }
]
}

```

Data Only

Dataonly is a modality like 3DS, but exclusive to the MasterCard brand. The purpose of the protocol is to decrease the fraud rate and increase approval rates in relation to a common transaction, as more data will be analyzed to support the issuer's decision making.

All DataOnly transactions are silently authenticated, just like a frictionless 3DS 2.0. This guarantees a smooth shopping experience for the user, but in contrast, it does not apply the liability shift benefit, that is, in case of chargebacks, the payment responsibility continues with the merchant, unlike 3DS authentication, which when successful, transfers this responsibility to the issuer.

In order to use the Product, it is necessary to contract it on the useRede portal, so that the Commerce can be activated in the MPI. Therefore, enable the service in the useRede portal, menu: para vender > e-commerce > 3DS > Contratar.

MPI (merchant plug-in) is the service that provides the establishment's integration with different issuers, in line with the certifications of the brands for processing the authentication.

e.Redre offers two ways to use the service, through MPI Rede or External MPI. The use of the MPI will be at the discretion of the establishment.

- Rede MPI: service already embedded in the e.Redre platform, without the need for additional contracting. In this scenario, the Network performs the authentication and authorization flow of the transaction.
- External MPI: service additionally contracted by the customer for integration with e.Redre, without influence of Rede in the authentication of the transaction. Therefore, in this scenario, Rede performs only the authorization flow.

In order to 3DS transactions to take place, issuers also need to be prepared to receive buyer authentication information. The main issuers in Brazil already offer this service to their customers.

Comparative table:

	Experience always without challenge	Influence on the issuer's approval decision	No transaction latency	Liability Shift
DataOnly	✓	✓	✓	✓

3DS	May be requested or not	✓		✓
-----	-------------------------	---	--	---

For more information, see the MasterCard infographic [here](#).

Data Only – Rede MPI

To use the Data Only mode with e.Redre's embedded MPI, simply add the *challengePreference* parameter to the Rede MPI request indicating the use of Data Only, see the table of "Request Parameters".

Request parameters:

POST /v1/transactions

Request:

```
{
  "capture": true,
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "2223000148400010",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "kind": "credit",
  "threeDSecure": {
    "embedded": true,
    "onFailure": "continue",
    "userAgent": "Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405",
    "device": {
      "dolorDepth": 1,
      "deviceType3ds": "BROWSER",
      "javaEnabled": false,
      "language": "BR",
      "screenHeight": 500,
      "screenWidth": 500,
      "timeZoneOffset": 3
    }
  },
  "challengePreference": "DATA_ONLY"
},
"urls": [
  {
    "kind": "threeDSecureSuccess",
    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/sucesso"
  }
],
}
```

```

{
  "kind": "threeDSecureFailure",
  "url": https://scommerce.userede.com.br/LojaTeste/Venda/erro
},
{
  "kind": " threeDSecureCallback",
  "url": "https://scommerce.userede.com.br/LojaTeste/Venda/callback"
}
]
}

```

Requisition parameters:

Name	Size	Type	Mandatory	Description
threeDSecure		threeDSecure	Yes	
threeDSecure /embedded		Boolean	No	<p>Informs if the MPI service used will be from Rede or a third party.</p> <ul style="list-style-type: none"> • true: uses Rede's MPI service • false: uses third-party's MPI service <p>Failure to send this field will be considered the use of Rede's MPI.</p>
threeDSecure /onFailure		Alphanumeric	No	<p>Defines how to proceed with the transaction if 3DS authentication is not successful.</p> <ul style="list-style-type: none"> • continue: proceeds with the financial transaction even if authentication fails • decline: do not proceed with the financial transaction if authentication fails <p>For debit transactions, the value of this parameter is automatically set to decline due to mandatory authentication.</p>

Name	Size	Type	Mandatory	Description
threeDSecure /userAgent	Up to 255	Alphanumeric	Yes	Browser identifier used by the buyer at the time of purchase.
threeDSecure /device				
threeDSecure /device/colorDepth	2	Numeric	Yes	Field representing the estimate of the color palette used for displaying images in bits per pixel. Obtained in the client's browser via the property.
threeDSecure /device/deviceType3ds	20	Alphanumeric	Yes	Field that indicates the type of device on which the authentication takes place.
threeDSecure /device/javaEnabled		Boolean	Yes	Boolean field representing the browser's ability to run Java. The value is the one returned by the navigator.javaEnabled property, true or false.
threeDSecure /device/language	10	Alphanumeric	Yes	Browser language in IETF BCP47 format, containing between 1 and 8 characters.
threeDSecure /device/screenHeight	6	Numeric	Yes	The total height of the customer's screen in pixels. The value is the one returned by the screen.height property.
threeDSecure /device/screenWidth	6	Numeric	Yes	The total width of the client's screen in pixels. The value is the one returned by the screen.width property.
urls		urls		
urls/kind		Alphanumeric	Yes	Field that identifies the type of the url. <ul style="list-style-type: none"> threeDSecureSuccess threeDSecureFailure threeDSecureCallback

Name	Size	Type	Mandatory	Description
urls/url	Up to 87	Alphanumeric	Yes	Field to inform the url that the buyer should be redirected after authentication and be notified via callback.
threeDSecure/challengePreference		Alphanumeric	No	Campo que indica a preferência de uso do Data Only. <ul style="list-style-type: none"> • DATA_ONLY

Attention: Data Only – Rede MPI only be used with the following messages:

- [Dynamic MCC](#): Messaging used for operations that transact using multiple merchant category codes;
- [Staged Digital Wallet Operators \(SDWO\)](#)
- [Rede brand tokenization](#)
- [External brand tokenization \(capture\)](#)

These messages can be used together or individually. See below an example of all messages together in a request:

```
{
  "capture": true,
  "kind": "credit",
  "reference": "pedido123",
  "amount": 2000,
  "cardholderName": "John Snow",
  "cardNumber": "2223000148400010",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "softDescriptor": "LOJADOZE",
  "tokenCryptogram": "ANbuvvxndbK2AAEShHMWGgADFA==",
  "paymentFacilitatorID": 22349202212,
  "independentSalesOrganizationID": 1234567,
  "subMerchant": {
    "mcc": 1234,
    "subMerchantID": 1234567890,
    "city": "Guarulhos",
    "cep": "07151-385",
    "state": "SP",
    "country": "BR",
  }
}
```

```

    "address": "Rua Pedro Luiz",
    "phoneNumber": "(11)91234-5678",
    "cnpj": "71.789.371/0001-42",
    "taxIdNumber": "99999999999"
  },
  "threeDSecure": {
    "embedded": true,
    "onFailure": "continue",
    "userAgent": "Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us)
AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405",
    "ipAddress": "192.168.130.20",
    "device": {
      "colorDepth": 1,
      "deviceType3ds": "BROWSER",
      "javaEnabled": false,
      "language": "BR",
      "screenHeight": 500,
      "screenWidth": 500,
      "timeZoneOffset": 3
    }
  },
  "challengePreference": "DATA_ONLY",
  "billing": {
    "address": "Rua Pedro Luiz",
    "city": "Guarulhos",
    "postalCode": "07151-385",
    "state": "SP",
    "country": "Brasil",
    "emailAddress": "email@user.com",
    "phoneNumber": "(11)91234-5678"
  }
},
"consumerBillPaymentService": {
  "businessApplicationIdentifier": "02"
},
"wallet": {
  "walletId": 1234567890,
  "processingType": "02",
  "senderTaxIdentification": "11122233344"
},
"urls": [
  {
    "kind": "threeDSecureSuccess",
    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/sucesso"
  },
  {
    "kind": "threeDSecureFailure",

```

```

    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/erro
  },
  {
    "kind": "threeDSecureCallback",
    "url": "https://scommerce.userede.com.br/LojaTeste/Venda/callback"
  }
]
}

```

Response:

```

{
  "reference": "pedido123",
  "tid": "8345000363484052380",
  "nsu": "663206341",
  "dateTime": "2017-02-28T08:54:00.000-03:00",
  "amount": 2099,
  "installments": 2,
  "cardBin": "544828",
  "last4": "0007",
  "returnCode": "00",
  "returnMessage": "Success.",
  "brand": {
    "name": "Mastercard.",
    "returnCode": "82",
    "returnMessage": "Policy (Mastercard use only)",
    "merchantAdviceCode": "03",
    "authorizationCode": "186376",
    "brandTid": "226332"
  },
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://sandbox-
eredede.useredecloud.com.br/v1/transactions/8345000363484052380"
    },
    {
      "method": "PUT",
      "rel": "capture",
      "href": "https://sandbox-
eredede.useredecloud.com.br/v1/transactions/8345000363484052380"
    },
    {

```

```

    "method": "POST",
    "rel": "refund",
    "href": "https://sandbox-
eredede.useredecloud.com.br/v1/transactions/8345000363484052380/refunds"
  }
]
}

```

Response parameters:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
tid	20	Alphanumeric	Unique transaction identifier number.
nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Datetime	Transaction date in the format YYYY-MM-DDThh:mm:ss.STZD .
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators. Examples: <ul style="list-style-type: none"> • R\$10.00 = 1000 • R\$0.50 = 50
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.

Data Only – External MPI

To use Data Only mode with an external MPI, the request will be the same as in 3DS 2.0. See the table of “Request Parameters”.

POST /v1/transactions

Request:

```
{
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "544828000000007",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "kind": "credit",
  "threeDSecure": {
    "embedded": false,
    "cavv": "BwABBylVaQAAAFwllVpAAAAAAA=",
    "eci": "04",
    "xid": "Nmp3VfDWMlEwZ05pWGN3SGo4TDA=",
    "threeDIndicator": "2",
    "directoryServerTransactionId": "f38e6948-5388-41a6-bca4-b49723c19437"
  }
}
```

Requisition parameters:

Name	Size	Type	Mandatory	Description
threeDSecure		threeDSecure	Yes	
threeDSecure /embedded		Boolean	No	<p>Informs if the MPI service used will be from Rede or a third party.</p> <ul style="list-style-type: none"> true: uses Rede's MPI service false: uses third-party's MPI service <p>Failure to send this field will be considered the use of Rede's MPI.</p>

Name	Size	Type	Mandatory	Description
threeDSecure / eci	2	Alphanumeric	Yes	Code returned to the MPI by the Brands that indicates the result of the bearer authentication with the Issuer. It must be sent only for the use of the 3DS authentication service. Debit transactions must be authenticated.
threeDSecure /cavv	Up to 32	Alphanumeric	Yes	Cryptogram code used to authenticate the transaction and sent by the establishment's MPI (may contain special characters). It must be sent only for the use of the 3DS authentication service.
threeDSecure /xid	28	Alphanumeric	*Yes	Authentication transaction ID added by MPI to the establishment (may contain special characters). It must be sent only for the use of the 3DS authentication service. Field used only for * Visa brand.
threeDSecure /threeDIndicator	1	Alphanumeric	Yes	3DS version used in the authentication process by MPI.
threeDSecure /directoryServerTransactionId	36	Alphanumeric	Yes	Authentication transaction ID added by MPI to the establishment (may contain special characters). It must be sent only for the use of the 3DS authentication service.

Response:

```
{
  "dateTime": "2019-08-01T10:01:03.000-03:00",
  "threeDSecure": {
    "embedded": true,
    "url":
    "https://scommerce.userede.com.br/adquirencia/mpi/auth?token=7z3nVtKnea4Gs4N%2bpeM2j8M%2fund%2bTfwDNVoAZLC7EqKC3gqz0j1DV0%2fpg928%2bh1sABmy1ZPlzyWxTqakpotWa%2f71q9wsDZ6b4Yk8KVXxR1xM00UmxR05V2bZiq%2bF%2fcrscUWE76js%2fDRokAa1RTEhn18Yw8Q7SAZafPc4YCpdWYgf"
```

```
VUxtZP4FDrC2KAqayjzPL85JjYDeujUdCgVhEhWimXLkkH6iJfC2qYppv94iHm4CYlAyKsxmjk76KFFOB
4uGocaHSR%2f0gDo0TyTxmxZ%2fTtH45WKq%2b4XmqG9LnDI%3d"
  },
  "returnCode": "220",
  "returnMessage": "Transaction request with authentication received. Redirect URL
sent."
}
```

Response parameters:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.
Reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Datetime	Transaction date in the format YYYY-MM-DDThh:mm:ss.sTZD .
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators. Examples: <ul style="list-style-type: none"> • R\$10.00 = 1000 • R\$0.50 = 50
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.

Table of ECIs

The ECI (Electronic Commerce Indicator) parameter is based on the value returned to the MPI by the Brands, which indicates the result of the bearer's authentication with the Issuer.

This, therefore, indicates the status of the authentication flow in a transaction, and whether the chargeback risk is transferred to the issuer or remains with the merchant. Check below the values used by the brands:

Brand	ECI	Meaning of the transaction	Chargeback risk
Elo	0	Unknown/ Not Specified/ Store does not participate in the program	Chargeback risk remains with the merchant
Elo	5	Cardholder Authenticated by the Issuer	Chargeback risk passes to the issuer
Elo	4	Transaction with In App authentication.	Used in Wallet transactions. Chargeback risk passes to the issuer
Elo	6	Cardholder Authentication Attempt by Acquirer Domain (authenticated by brand)	Chargeback risk passes to the issuer
Elo	7	Unauthenticated eCommerce Transaction	Chargeback risk remains with the merchant
Mastercard	0	Incomplete or failed authentication attempt	Chargeback risk remains with the merchant
Mastercard	1	Mastercard Stand-In Authentication	Chargeback risk passes to the issuer
Mastercard	2	Successful Authentication	Chargeback risk passes to the issuer
Mastercard	4	Data Only authentication successful	Chargeback risk remains with the merchant
Mastercard	7	Recurring	Chargeback risk remains with the merchant
Visa	5	Card authentication successful	Chargeback risk passes to the issuer
Visa	6	Authentication was attempted but did not or could not complete; possible reasons, and the card or its Issuing Bank does not yet participate. (authenticated by brand)	Chargeback risk passes to the issuer
Visa	7	This ECI is returned in two scenarios: - 3DS: Authentication was not successful or was not attempted.	In both cases, the risk of chargeback remains with the merchant

Brand	ECI	Meaning of the transaction	Chargeback risk
		- Data Only: Data Only authentication successful	

Zero Dollar

The Zero Dollar transaction allows for prior validation to see if the cardholder's card and data sent before processing the transaction are valid. This type of transaction does not generate any type of charge for the buyer, avoiding undue debits on their balance.

The service is available for Visa, MasterCard, Elo, Hiper and Hipercard on credit. On debit, the service is available for Visa, Mastercard and Elo brand. Zero Dollar is mandatory when you intend to store the card, while for other operations it is highly recommended in order to validate the card before starting the standard transactional flow.

The securityCode parameter will be mandatory for Zero Dollar validations in all brands.

To enable the service, access the useRede portal, menu: *para vender > e-commerce > Zero Dollar > Contratar*. After confirming the activation of the service, Zero Dollar transactions must be sent as authorization with automatic capture (**capture = true**), **informing the value 0 in the amount parameter**.

IMPORTANT:

- This type of transaction cannot be canceled.
- This type of transaction must not be submitted as recurring (subscription=true). First perform the Zero Dollar validation following the parameters specified below and then it will be possible to use the card for recurring or non-recurring transactions.
- For Visa and Mastercard transactions, within the brand field group the ABECS code of the successful brands is returned as 85. For more details on the Rede codes in this case, see [Card Center Returns](#).

POST /v1/transactions

Request

```
{
  "capture": true,
  "kind": "credit",
```

```

"reference": "TZD001",
"amount": 0,
"cardholderName": "John Snow",
"cardNumber": "5448280000000007",
"expirationMonth": 12,
"expirationYear": 2028,
"securityCode": "235"
}

```

Requisition parameters:

Name	Size	Type	Mandatory	Description
capture		Boolean	No	Defines whether the transaction will be captured automatically or later. Failure to send this field will be considered automatic capture (true). For debit and Zero Dollar transactions, when this field is sent, the parameter must be set as true, indicating automatic capture. For debit and Zero Dollar transactions, when this field is sent, the parameter must be set as true, indicating automatic capture.
kind		credit / debit	No	Type of transaction to be performed. Failure to submit this field will be considered a credit.
reference	Up to 16	Alphanumeric	Yes	Order code generated by the establishment.
amount	Up to 10	Numeric	Yes	For Zero Dollar transaction send the value 0.
cardholderName	Up to 30	Alphanumeric	No	Cardholder's name. Do not send special characters
cardNumber	Up to 19	Alphanumeric	Yes	Card number.
expirationMonth	Up to 2	Numeric	Yes	Card expiration month. From 1 to 12.
expirationYear	2 or 4	Numeric	Yes	Card expiration year. Example: 2028 or 28.

Name	Size	Type	Mandatory	Description
securityCode	Up to 4	Alphanumeric	Yes	The card security code is usually located on the back of the card. Sending this parameter guarantees a greater possibility of approval of the transaction.

Response:

```
{
  "reference": "010224113900",
  "tid": "10402402011022100032",
  "nsu": "500022009",
  "dateTime": "2024-02-01T11:39:09-03:00",
  "amount": 60000,
  "installments": 4,
  "cardBin": "544828",
  "last4": "0007",
  "brand": {
    "name": "Mastercard",
    "returnCode": "00",
    "returnMessage": "Success.",
    "authorizationCode": "28943A",
    "brandTid": "MCS2405664610201"
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://api-hom.userede.com.br/erede/v1/transactions/10402402011022100032"
    },
    {
      "method": "POST",
      "rel": "refund",
      "href": "https://api-hom.userede.com.br/erede/v1/transactions/10402402011022100032/refunds"
    }
  ]
}
```

Response parameters:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	<ul style="list-style-type: none"> Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh: mm: ss.STZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.

Cancellation

Cancellation can be requested for all transactions, as instructed below:

- Authorization

The authorization cancellation operation (without automatic capture) is only allowed for the total cancellation of the transaction and must be requested within the stipulated period for each branch. After this period, the authorization is canceled automatically.

- Capture and authorization with automatic capture

The operation of canceling the capture and authorization with automatic capture can be carried out partially or totally through the available channels.

In full cancellation, the transaction will have the status "Canceled", while in partial cancellation, the status will be kept as "Approved" until the transaction is canceled in full.

Cancellation requests can be made within 7 days for debit transactions and for credit transactions the standard is up to 90 days, but it may vary depending on the field of activity of each establishment.

For cancellations requested on the same day as the authorization transaction or authorization with automatic capture, processing will be carried out immediately, otherwise, processing will be carried out on D+1.

Attention: Every cancellation request sent after 9:30 PM are processed on the next day.

We remind you that for Maestro transactions (debit), it is possible to carry out only a partial cancellation. This is a Mastercard brand rule, which can send confirmation/processing of this cancellation within 5 working days.

Cancellation can be requested for all transactions.

To support our customers in better identifying cancellation returns denied by brand rules, since January 29th, 2023 it is possible to receive two new cancellation scenarios through the e.Rede API. Initially, they will be included in existing returns (351 and 354) and you must prepare to receive the definitive returns from May 01st,2023.

For the Mastercard brand, in which more than **one partial cancellation of the debit is not allowed**, until May 01st,2023, you will receive this return through code 355, which already exists.

For Mastercard debit **disputed chargeback transactions**, you will receive code 351 by May 1st,2023, also existing.

After this period, the new returns are as shown in the table below:

Until May 1 st , 2023– returns that will include the new scenarios provisionally		From May 1 st ,2023 – definitive returns	
355	Transaction already canceled	373	No further Refund allowed
351	Forbidden	374	Refund not allowed. Chargeback requested

See [Cancellation returns](#) for more details.

To test the scenarios see Sandbox Tutorial > [Simulate errors](#).

POST /v1/transactions/{tid}/refunds

Parameters

Authorization Service Access Token: Basic {{hash_pv_token}}

***required**

string

(header)

Example : Basic NzM4NTQ5Njc6NjA2OWEwMjZjZjQ1NDcwNjk5MGE4MDFhYjVmZThlMzY=

tid *required Unique transaction identifier number. Maximum Size (20).

string

(path)

Example : 9274256037511432483

Cancellation can be requested for all transactions.

Request:

```
{
  "amount": 2000,
  "urls": [
    {
      "kind": "callback",
      "url": "https://cliente.callback.com.br"
    }
  ]
}
```

Requisition parameters:

Name	Size	Type	Mandatory	Description
amount	Up to 10	Numeric	Yes	Cancellation value without thousands and decimal separator. Examples: R\$ 10.00 = 1000 R\$ 0.50 = 50
urls		urls	No	
urls/kind		Alphanumeric	No	Field that identifies the type of url: callback.
urls/url	Up to 500	Alphanumeric	No	Url that will receive the callback with the cancellation status after being processed by Rede. It is also possible to

Name	Size	Type	Mandatory	Description
				register the url in the userede portal. Click here for more information

Response:

```
{
  "returnCode": "360",
  "returnMessage": "Refund request has been successful",
  "refundId": "d21c0fa9-aa0f-4b9f-aedc-a1d4ed08e03d",
  "tid": "9274256037511432483",
  "nsu": "750004939",
  "refundDateTime": "2017-02-11T08:45:00.000-03:00",
  "cancelId": "786524681",
  "links": [
    {
      "method": "GET",
      "rel": "refund",
      "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/9274256037511432483/refunds/5938e353-a6e7-
430f-bac3-869acf1e7665"
    },
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/9274256037511432483"
    },
    {
      "method": "GET",
      "rel": "refunds",
      "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/9274256037511432483/refunds"
    }
  ]
}
```

Response parameters:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code (see cancellation returns table).
returnMessage	Up to 256	Alphanumeric	Transaction return message (see cancellation returns table).
refundId	36	Alphanumeric	Cancellation request return code generated by Rede. If the transaction is canceled through a channel other than the API, this field will return empty.
tid	20	Alphanumeric	Unique transaction identifier number.
nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
refundDateTime		Datetime	Cancellation date in YYYY-MM-DDThh:mm:ss.sTZD format.
cancelId	Up to 15	Alphanumeric	Transaction identifier code of cancellation request returned only in D+1 requests.

Notification URL

The notifications URL (callback) allows the data of a transaction to be returned via POST after processing the cancellations performed on D+1. The URL can be informed in the API itself or by accessing the Rede portal at: *para vender > e-commerce > notificação automática*. We emphasize that if the URL is informed in both channels, the priority for sending notifications will always be the one informed in the API.

IMPORTANT: In line with market practices to ensure greater safety, update your TLS 1.2 compliant public certificate. As of June 29th, 2018, previous versions such as 1.1 and 1.0 will no longer work.

After informing the URL that will receive the notification, the information will be returned in the following format:

Name	Size	Type	Description
type		Alphanumeric	Event type used for transaction: refund .
tid	20	Alphanumeric	Unique transaction identifier number.
nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
date		Datetime	Cancellation date in YYYY-MM-DDThh:mm:ss.sTZD format.

Name	Size	Type	Description
amount	Up to 10	Alphanumeric	Cancellation value.
status	Up to 10	Alphanumeric	<p>? Done (Cancellation completed)</p> <p>? Denied (Cancellation denied)</p> <p>? Processing(Cancellation in process)</p>
cancellationNotice	Up to 15	Alphanumeric	Cancellation request transaction identifier code (cancelld).
refundId	36	Alphanumeric	Cancellation request return code generated by Rede. If the transaction is canceled through a channel other than the API, this field will return empty.

Transaction query

Transaction query can be performed in two ways. The first is informing the tid generated in the authorization transaction. The second is informing the order number created by the establishment (reference).

Note: The deadline for querying pending pre-authorizations and zero dollar transactions is 60 days. After this period, the query status will return as: not found.

Query by tid

The transaction query can be performed in two ways:

- The first is informing the tid generated in the authorization transaction. [this endpoint]
- The second is informing the order number created by the establishment (reference).[endpoint above].

Note: The deadline for querying pending pre-authorizations and zero dollar transactions is 60 days. After this period, the query status will return as: not found.

GET /v1/transactions

```
{
  "requestDateTime": "2017-03-12T08:54:00.000-03:00",
  "authorization": {
    "dateTime": "2017-03-11T08:54:00.000-03:00",
    "returnCode": "00",
    "returnMessage": "Success."
  }
}
```

```

"affiliation": 37502603,
"status": "Pending",
"reference": "pedido123",
"tid": "8345000363484052380",
"nsu": "663206341",
"authorizationCode": "186376",
"kind": "Credit",
"amount": 2000,
"installments": 2,
"currency": "BRL",
"cardHolderName": "John Snow",
"cardBin": "544828",
"last4": "0007",
"softDescriptor": "lojarede",
"origin": 1,
"subscription": false,
"distributorAffiliation": 0,
"brand": {
  "name": "Mastercard",
  "returnCode": "00",
  "returnMessage": "Success.",
  "authorizationCode": "110475",
  "brandTid": "MCS1526050620328"
},
},
"capture": {
  "dateTime": "stringstringstringstringstrin",
  "nsu": "string",
  "amount": 0
},
"threeDSecure": {
  "embedded": true,
  "eci": "st",
  "cavv": "BwABBylVaQAAAAFwllVpAAAAA=",
  "xid": "stringstringstringstringstri",
  "returnCode": "str",
  "returnMessage": "string"
},
"refunds": {
  "dateTime": "stringstringstringstringstrin",
  "refundId": "stringstringstringstringstringstring",
  "status": "Processing",
  "amount": 0
},
"links": [
  {

```

```

    "method": "POST",
    "rel": "refund",
    "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/8345000363484052380/refunds"
  },
  {
    "method": "PUT",
    "rel": "capture",
    "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/8345000363484052380"
  }
]
}

```

Parameters

Authorization

Service Access Token: Basic {{hash_pv_token}}

***required**

string

Example : Basic

NzM4NTQ5Njc6NjA2OWEwMjZjZjQ1NDcwNjk5MGE4MDFhYjVmZThlMzY=

(header)

tid ***required**

Unique transaction identifier number. Maximum Size (20).

string

Example : 9274256037511432483

(path)

Requisition parameters:

Name	Size	Type	Mandatory	Description
tid	20	Alphanumeric	Yes	Unique transaction identifier number.

Response parameters:

Name	Size	Type	Description
requestDateTime		Datetime	Request date in YYYY-MM-DDThh:mm:ss.STZD format.
Authorization		authorization	

Name	Size	Type	Description
authorization/dateTime		Datetime	Authorization transaction date in YYYY-MM-DDThh:mm:ss.STZD format.
authorization/returnCode	Up to 3	Alphanumeric	Transaction return code.
authorization/returnMessage	Up to 256	Alphanumeric	Transaction return message.
authorization/affiliation	Up to 9	Numeric	Establishment's affiliation number (PV).
authorization/status		Alphanumeric	Transaction status: <ul style="list-style-type: none"> • Approved • Denied • Canceled • Pending
authorization/reference	Up to 16	Alphanumeric	Order number generated by the establishment.
authorization/tid	20	Alphanumeric	Unique transaction identifier number.
authorization/nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorization/authorizationCode	6	Alphanumeric	Transaction Authorization Number returned by the card issuer.
authorization/kind	Up to 10	Alphanumeric	Payment method used in the original transaction (Credit or Debit).
authorization/amount	Up to 10	Numeric	Total purchase amount without thousands separator. Examples: ☐ 1000 = R\$10.00 ☐ R\$ 0.50 = 50
authorization/installments	Up to 2	Numeric	Number of installments.

Name	Size	Type	Description
authorization/cardHolderName	Up to 30	Alphanumeric	Cardholder's name printed on the card.
authorization/cardBin	6	Alphanumeric	6 first digits of the card.
authorization/last4	4	Alphanumeric	4 last digits of the card.
authorization/softDescriptor	Up to 18*	Alphanumeric	Message that will be displayed next to the name of the establishment on the cardholder's invoice.
authorization/origin	Up to 2	Numeric	Identifies the source of the transaction. <ul style="list-style-type: none"> e.Redre - 1
authorization/subscription		Boolean	Informs the issuer if the transaction comes from a recurrence. If the transaction is a recurrence, send true . Otherwise, send false . Failure to submit this field will be considered the value false. Rede does not manage recurrence schedules, it only allows establishments to indicate whether the transaction originated from a recurrence.
authorization/distributorAffiliation	Up to 9	Numeric	Distributor's affiliation number (PV).
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional

Name	Size	Type	Description
			information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones. For more details see the section Recurrence and Card-on-file
capture		capture	
capture/dateTime		Datetime	Capture transaction date in YYYY-MM-DDThh:mm:ss.sTZD format.
capture/nsu	Up to 12	Alphanumeric	Sequential number returned by Rede in the capture transaction.
capture/amount	Up to 10	Numeric	Capture value.
threeDSecure		threeDSecure	
threeDSecure/embedded		Boolean	Informs if the MPI service used will be from Rede or a third party.
threeDSecure/eci	2	Alphanumeric	Code returned to MPI by Brands that indicate the result of the cardholder authentication with the Issuer. It must be sent only to use the 3DS authentication service. Debit transactions must be authenticated.
threeDSecure/cavv	Up to 32	Alphanumeric	Cryptogram code used to authenticate the transaction and sent by the establishment's MPI (may contain special characters). It must be sent only for the use of the 3DS authentication service.

Name	Size	Type	Description
threeDSecure/xid	28	Alphanumeric	Authentication transaction ID sent by MPI to the establishment (may contain special characters). It must be sent only for the use of the 3DS authentication service. Field used only for Visa brand.
threeDSecure/returnCode	3	Alphanumeric	Transaction with 3ds return code.
threeDSecure/returnMessage	Up to 256	Alphanumeric	Transaction with 3ds return message
refunds		refunds	
refunds/dateTime		Datetime	Cancellation transaction date in YYYY-MM-DDThh:mm:ss.STZD format.
refunds/refundId	36	Alphanumeric	Cancellation request return code generated by Rede.
refunds/status	Up to 10	Alphanumeric	Status of the cancellation request. <ul style="list-style-type: none"> • Done (Cancellation completed) • Denied (Cancellation denied) • Processing (Cancellation in process)
refunds/amount	Up to 10	Numeric	Cancellation value.

Query by order code (reference)

The transaction query can be performed in two ways:

- The first is informing the tid generated in the authorization transaction. [endpoint below]
- The second is informing the order number created by the establishment (reference). [this endpoint]

Note: The deadline for querying pending pre-authorizations and zero dollar transactions is 60 days. After this period, the query status will return as: not found.

GET /v1/transactions

Parameters

Authorization **required* Service Access Token: Basic {{hash_pv_token}}

string *Example* : Basic
 (header) NzM4NTQ5Njc6NjA2OWEwMjZjZjQ1NDcwNjk5MGE4MDFhYjVmZThlMzY=

reference **required* Order number generated by the establishment. Maximum Size (16).

string *Example*: order123
 (query)

Requisition parameters:

Name	Size	Type	Mandatory	Description
reference	Up to 16	Alphanumeric	Yes	Order number generated by the establishment.

Response:

```
{
  "requestDateTime": "2017-03-12T08:54:00.000-03:00",
  "authorization": {
    "dateTime": "2017-03-11T08:54:00.000-03:00",
    "returnCode": "00",
    "returnMessage": "Success.",
    "affiliation": 37502603,
    "status": "Pending",
    "reference": "pedido123",
    "tid": "8345000363484052380",
    "nsu": "663206341",
    "authorizationCode": "186376",
    "kind": "credit",
    "amount": 2099,
    "installments": 2,
    "currency": "BRL",
    "cardholderName": "John Snow",
    "cardBin": "544828",
    "last4": "0007",
    "softDescriptor": "lojarede",
    "origin": 1,
    "subscription": false,
    "distributorAffiliation": 0
  },
  "capture": {
```

```

    "dateTime": "stringstringstringstringstrin",
    "nsu": "string",
    "amount": 0
  },
  "threeDSecure": {
    "embedded": true,
    "eci": "st",
    "cavv": "BwABBylVaQAAAAFwllVpAAAAAAA=",
    "xid": "stringstringstringstringstri",
    "returnCode": "str",
    "returnMessage": "string"
  },
  "refunds": {
    "dateTime": "stringstringstringstringstrin",
    "refundId": "stringstringstringstringstringstringstring",
    "status": "Processing",
    "amount": 0
  },
  "links": [
    {
      "method": "POST",
      "rel": "refund",
      "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/8345000363484052380/refunds"
    },
    {
      "method": "PUT",
      "rel": "capture",
      "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/8345000363484052380"
    }
  ]
}

```

Response parameters:

Name	Size	Type	Description
requestDateTime		Datetime	Request date in YYYY-MM-DDThh:mm:ss.sTZD format.
authorization		authorization	

Name	Size	Type	Description
authorization/dateTime		Datetime	Authorization transaction date in YYYY-MM-DDThh:mm:ss.sTZD format.
authorization/returnCode	Up to 3	Alphanumeric	Transaction return code.
authorization/returnMessage	Up to 256	Alphanumeric	Transaction return message.
authorization/affiliation	Up to 9	Numeric	Establishment's affiliation number (PV).
authorization/status		Alphanumeric	Transaction status: <ul style="list-style-type: none"> • Approved • Denied • Canceled • Pending
authorization/reference	Up to 16	Alphanumeric	Order number generated by the establishment.
authorization/tid	20	Alphanumeric	Unique transaction identifier number.
authorization/nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorization/authorizationCode	6	Alphanumeric	Transaction Authorization Number returned by the card issuer.
authorization/kind	Up to 10	Alphanumeric	Payment method used in the original transaction (Credit or Debit).
authorization/amount	Up to 10	Numeric	Total purchase amount without thousands separator. Examples: ☐ 1000 = R\$10.00 ☐ R\$ 0.50 = 50

Name	Size	Type	Description
authorization/installments	Up to 2	Numeric	Number of installments.
authorization/cardHolderName	Up to 30	Alphanumeric	Cardholder's name printed on the card.
authorization/cardBin	6	Alphanumeric	6 first digits of the card.
authorization/last4	4	Alphanumeric	4 last digits of the card.
authorization/softDescriptor	Up to 18*	Alphanumeric	Message that will be displayed next to the name of the establishment on the cardholder's invoice.
authorization/origin	Up to 2	Numeric	Identifies the source of the transaction. <ul style="list-style-type: none"> e.Redre - 1
authorization/subscription		Boolean	<p>Informs the issuer if the transaction comes from a recurrence. If the transaction is a recurrence, send true. Otherwise, send false.</p> <p>Failure to submit this field will be considered the value false.</p> <p>Rede does not manage recurrence schedules, it only allows establishments to indicate whether the transaction originated from a recurrence.</p>
authorization/distributorAffiliation	Up to 9	Numeric	Distributor's affiliation number (PV).
Capture		capture	
capture/dateTime		Datetime	Capture transaction date in YYYY-MM-DDThh:mm:ss.sTZD format.
capture/nsu	Up to 12	Alphanumeric	Sequential number returned by Rede in the capture transaction.
capture/amount	Up to 10	Numeric	Capture value.

Name	Size	Type	Description
threeDSecure		threeDSecure	
threeDSecure/embedded		Boolean	Informs if the MPI service used will be from Rede or a third party.
threeDSecure/eci	2	Alphanumeric	Code returned to MPI by Brands that indicate the result of the cardholder authentication with the Issuer. It must be sent only to use the 3DS authentication service. Debit transactions must be authenticated.
threeDSecure/cavv	Up to 32	Alphanumeric	Cryptogram code used to authenticate the transaction and sent by the establishment's MPI (may contain special characters). It must be sent only for the use of the 3DS authentication service.
threeDSecure/xid	28	Alphanumeric	Authentication transaction ID sent by MPI to the establishment (may contain special characters). It must be sent only for the use of the 3DS authentication service. Field used only for Visa brand.
threeDSecure/returnCode	3	Alphanumeric	Transaction with 3ds return code.
threeDSecure/returnMessage	Up to 256	Alphanumeric	Transaction with 3ds return message
Refunds		refunds	
refunds/dateTime		Datetime	Cancellation transaction date in YYYY-MM-DDThh:mm:ss.sTZD format.
refunds/refundId	36	Alphanumeric	Cancellation request return code generated by Rede.

Name	Size	Type	Description
refunds/status	Up to 10	Alphanumeric	Status of the cancellation request. <ul style="list-style-type: none"> • Done (Cancellation completed) • Denied (Cancellation denied) • Processing (Cancellation in process)
refunds/amount	Up to 10	Numeric	Cancellation value.

Cancellation Query

It is used to query cancellation information from a request sent, being possible to query informing the TID for a more detailed consultation, and the refundId for a query of a specific cancellation.

The query of the final status of the cancellation can be carried out through the transaction query API, the Rede portal, or the electronic statement one day after the cancellation request has been made.

Cancellation query by tid

Request

The cancellation query by tid lists all the cancellations that were performed on the transaction.

GET /v1/transactions

Parameters

Authorization **required* Service Access Token: Basic {{hash_pv_token}}

string

Example : Basic

(header)

NzM4NTQ5Njc6NjA2OWEwMjZjZjQ1NDcwNjk5MGE4MDFhYjVmZThlMzY=

tid *required Unique transaction identifier number. Maximum Size (20).

string *Example : 9274256037511432483*

(path)

Requisition parameters:

Name	Size	Type	Mandatory	Description
tid	20	Alphanumeric	Yes	Unique transaction identifier number.

Response:

```
{
  "refunds": [
    {
      "refundId": "d21c0fa9-aa0f-4b9f-aedc-a1d4ed08e03d",
      "refundDateTime": "2017-02-23T00:00:00.000-03:00",
      "cancelId": "917054151307902",
      "status": "Processing",
      "amount": 1000
    }
  ],
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://sandbox-erede.useredecloud.com.br/v1/transactions/23457040715"
    }
  ]
}
```

Response parameters:

Name	Size	Type	Description
refundId	36	Alphanumeric	Cancellation return code generated by Rede. If the transaction is canceled through a channel other than the API, this field will return empty.

Name	Size	Type	Description
refundDateTime		Datetime	Cancellation date in YYYY-MM-DDThh:mm:ss.sTZD format.
cancelId	Up to 15	Alphanumeric	Transaction identifier code of cancellation request returned only in D+1 requests.
status	Up to 10	Alphanumeric	Status of cancellation requests <ul style="list-style-type: none"> • Done (Cancellation completed) • Denied (Cancellation denied) • Processing (Cancellation in process)
amount	Up to 10	Numeric	Cancellation value without thousands and decimal separator.

Cancellation query by refundId

The refundId cancellation query lists a specific cancellation request.

The refundId cancellation query lists a specific cancellation request.

GET /v1/transactions

Parameters

Authorization *required Service Access Token: Basic {{hash_pv_token}}

string (header) *Example* : Basic NzM4NTQ5Njc6NjA2OWEwMjZjZjQ1NDcwNjk5MGE4MDFhYjVmZThlMzY=

tid *required Unique transaction identifier number. Maximum Size (20).

string (path) *Example* : 9274256037511432483

refundId *required Cancellation return code generated by Rede. If the transaction is canceled through a channel other than the API, this field will return empty, and it is not possible to query by refundId. Exact Size (36).

string (path) *Example* : d21c0fa9-aa0f-4b9f-aedc-a1d4ed08e03d

Response:

```
{
  "refundId": "d21c0fa9-aa0f-4b9f-aedc-a1d4ed08e03d",
  "tid": "9274256037511432483",
  "refundDateTime": "2017-01-28T08:33:00.000-03:00",
  "cancelId": "786524681",
  "amount": 2000,
  "status": "Processing",
  "statusHistory": [
    {
      "status": "Processing",
      "dateTime": "2017-01-28T08:33:00.000-03:00"
    }
  ],
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://sandbox-
eredede.userredecloud.com.br/v1/transactions/9274256037511432483."
    }
  ]
}
```

Response parameters:

Name	Size	Type	Mandatory
refundId	36	Alphanumeric	Cancellation return code generated by Rede. If the transaction is canceled through a channel other than the API, this field will return empty, and it is not possible to query by refundId.
tid	20	Alphanumeric	Unique transaction identifier number
refundDateTime		Datetime	Date of cancellation in YYYY-MM-DDThh:mm:ss.sTZD format.
cancelId	Up to 15	Alphanumeric	Transaction identifier code of cancellation request returned only in D+1 requests.

Name	Size	Type	Mandatory
amount	Up to 10	Numeric	Cancellation value without thousands and decimal separator.
statusHistory		statusHistory	
statusHistory/status	Up to 10	Alphanumeric	Status history of cancellation requests <ul style="list-style-type: none"> • Done (Cancellation completed) • Denied (Cancellation denied) • Processing (Cancellation in process)
statusHistory/dateTime		Datetime	Date of cancellation request in YYYY-MM-DDThh:mm:ss.sTZD format.
returnCode	Up to 3	Alphanumeric	Transaction return code
returnMessage	Up to 256	Alphanumeric	Transaction return message

softDescriptor

The identification on the invoice (SoftDescriptor or DBA) is a parameter that helps the cardholder to identify the transaction generated on the card invoice.

The parameter consists of 22 characters. The SoftDescriptor is divided into two parts, in which the first part is registered on the Rede portal and we call it a hard descriptor, as it is unique per transaction of that PV. The second part is dynamic, and is sent to each transaction request via API, this part is what we call SoftDescriptor.

These values are imputed in the capture message for the brand, and separated with an * (asterisk).

The hard descriptor can have a maximum of 12 characters and it will be variable depending on the amount of SoftDescriptor characters that come in the request. That is, the hard descriptor is registered only once, and the way it appears on the invoice varies according to the size of the SoftDescriptor, below we exemplify the e.Redes API rules for combining both fields:

- If between 1 and 9 positions are sent in the request in the SoftDescriptor, the composition on the invoice will be 12 hard characters + 1 to 9 Soft characters, with the asterisk, totaling the 22 characters open for this information.

Example: Assuming that you are registered on the portal as hard descriptor “REDECOMMERCE” and SoftDescriptor “PRODUTO01”, the final customer's invoice will show REDECOMMERCE*PRODUTO01

Example with spaces in the SoftDescriptor: Assuming that it is registered in the portal as hard descriptor “REDECOMMERCE” and SoftDescriptor “PRODU”, the final customer's invoice will show REDECOMMERCE*PRODU

- If between 10 and 14 positions are sent in the request in the SoftDescriptor, the composition on the invoice will be 7 characters of hard + 10 to 14 characters of Soft, with the asterisk, totaling the 22 characters open for this information.

Example: Assuming that you are registered on the portal as hard descriptor “REDECOMMERCE” and SoftDescriptor “PRODUTODIGIT01”, the final customer's invoice will show REDECOM*PRODUTODIGIT01

Example with spaces in the SoftDescriptor: Assuming that it is registered in the portal as hard descriptor “REDECOMMERCE” and SoftDescriptor “PRODU”, the final customer's invoice will show REDECOM*PRODU

- If between 15 and 18 positions are sent in the request in the SoftDescriptor, the composition on the invoice will be 3 characters from the hard + 15 to 18 from the Soft, with the asterisk, totaling the 22 characters open for this information.

Example: Assuming that you are registered on the portal as hard descriptor “REDECOMMERCE” and SoftDescriptor “PRODUTODIGITAL0001”, the final customer's invoice will show RED*PRODUTODIGITAL0001

Example with spaces in the SoftDescriptor: Assuming that it is registered in the portal as hard descriptor “REDECOMMERCE” and SoftDescriptor “PRODU ”, the final customer's invoice will show RED*PRODU

Important: If you use MPI Rede, do not use spaces or special characters in the SoftDescriptor, as this will result in errors in the authentication of the transaction.

To use this feature, access the Rede portal in the menu *para vender > e-commerce > Identificação na fatura*, or contact the Rede Call Center. If the name is not registered, the service will not be enabled.

After enabling the service via the portal, the functionality will be available within 24 hours.

The parameter must be sent with the request for credit (authorization or authorization with automatic capture) or debit transactions.

Dynamic MCC

The merchant's category code, known as MCC, sent by the marketplace or facilitator, can be dynamic according to the information of the merchant that is making the transaction.

For this scenario, it is mandatory to send the softdescriptor. [Click here](#) for more information.

For dynamic MCC information, the name registered in the portal, menu *para vender > e-commerce > Identificação na fatura*, is equivalent to the name of the facilitator (sub accreditor).

POST /v1/transactions

Request

```
{
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "softDescriptor": "LOJADOZE",
  "PaymentFacilitatorID": 22349202212,
  "IndependentSalesOrganizationID": 1234567,
  "subMerchant": {
    "mcc": "1111",
    "subMerchantID": 1234567890,
    "address": "Rua Acre",
    "city": "CAPIVARI",
    "state": "SP",
    "country": "BRA",
    "zipcode": "07064-010",
    "tin": "71.789.371/0001-42",
    "taxIdNumber": "99999999999999"
  }
}
```

Requisition parameters:

Name	Size	Type	Mandatory	Description
softDescriptor	Up to 18*	Alphanumeric	Yes*	Personalized phrase that will be printed on the cardholder's invoice.
paymentFacilitatorID	Up to 11	Numeric	Yes*	Facilitator code.
independentSalesOrganizationID	Up to 11	Numeric	No	Independent sales organization code.

Name	Size	Type	Mandatory	Description
subMerchant		subMerchant		
subMerchant / mcc	4	Alphanumeric	Yes*	MCC of the sub shopkeeper.
subMerchant / subMerchantID	Up to 15	Alphanumeric	Yes*	Sub shopkeeper's Code
subMerchant / address	Up to 48	Alphanumeric ¹	No*	Sub shopkeeper's Address
subMerchant / city	Up to 13	Alphanumeric ¹	No*	Sub shopkeeper's City
subMerchant / state	2	Alphabetical	Yes*	Sub shopkeeper's State
subMerchant / country	Up to 3	Alphanumeric	Yes*	Sub shopkeeper's Country
subMerchant / zipcode	Up to 9	Alphanumeric	Yes*	Sub shopkeeper's Postal code
subMerchant / cnpj	Up to 18	Numeric	No*	Sub shopkeeper's CNPJ (TIN)
subMerchant / taxIdNumber	Up to 14	Alphanumeric	Yes*	Sub shopkeeper's CNPJ (TIN) or CPF (ITIN)

***Mandatory fields for Subacquirers and Marketplaces operation, for more details see the specific section [here](#).**

Note that request parameters inside the “Submerchant” group must always start with a lowercase letter.

*The hard descriptor can have a maximum of 12 characters and it will be variable depending on the amount of SoftDescriptor characters that come in the request. In other words, **the hard descriptor is registered only once**, and the way it appears on the invoice **varies according to the SoftDescriptor size**. Up to 18 positions in the SoftDescriptor can be sent in the request and in this case, the composition on the invoice will be 3 characters of hard + 18 of Soft, with the asterisk, totaling the 22 characters open for this information.

Important: Due to the LGPD (General Data Protection Law), the following fields of the “SubMerchant” key: SubMerchantID, Address, City, State, Country, Zip Code and CNPJ (TIN), even when sent in the request, are not returned in transaction queries.

Attention:

- To ensure proper processing of the transaction, special characters must not be included.

Response

```

{
  "reference": "160224094727",
  "tid": "10012402160947499092",
  "nsu": "557854383",
  "dateTime": "2024-02-16T09:47:49-03:00",
  "amount": 1000,
  "cardBin": "544828",
  "last4": "0007",
  "brand": {
    "name": "Mastercard",
    "returnMessage": "Success.",
    "returnCode": "00",
    "brandTid": "263942",
    "authorizationCode": "263942"
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/10012402160947499092"
    },
    {
      "method": "POST",
      "rel": "refund",
      "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/10012402160947499092/refunds"
    }
  ]
}

```

Response parameters:

Name	Size	Type	Description
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.

Name	Size	Type	Description
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh: mm: ss.sTZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones. For more details see the section Recurrence and Card-on-file
returnCode	Up to 3	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.

Subacquirers and Marketplaces

Public

Sub-acquirers: company integrated with an acquirer that enables other companies or individuals to accept card payments by intermediating the financial flow of transactions.

Marketplace: e-commerce that sells third-party products, operating as a virtual shopping center. It is subject to the same rules as a Sub-Acquirer as it mediates the financial flow for the Seller;

Transactional message

Circular 3978 requires Subacquirers and Marketplace to identify the final beneficiaries at the time of the transaction. To comply with this standard, it is mandatory to send identifying fields in the transaction message, as per the guidelines below:

- **Softdescriptor:** It is a parameter that helps the cardholder to identify the transaction generated on the card invoice.

This parameter is made up of two parts, the first is the **Hard Descriptor**, which is unique to the subacquirer, and the second is dynamic, which we call **Softdescriptor**, it identifies the subestablishment of the transaction.

This field is mandatory and must be up to **22 characters** long. The Hard Descriptor can have a maximum of 12 characters and it will vary depending on the number of SoftDescriptor characters that appear in the request.

*The hard descriptor can have a maximum of 12 characters and it will vary depending on the number of SoftDescriptor characters that appear in the request. In other words, **the hard descriptor is registered once, and the way it appears on the invoice varies according to the size of the SoftDescriptor**. Up to 18 positions in the SoftDescriptor can be sent in the request and in this case, the composition on the invoice will be 3 characters of hard + 18 of Soft, with the asterisk, totaling the 22 characters open for this information.

For more details on sending this field, click here or access the session [Softdescriptor](#);

- **PFID (paymentFacilitatorID):** Payment Facilitator Code in each brand
- **SubmerchantID:** This code is generated by the Payment facilitators **subMerchant / address:** Endereço do Subestabelecimento (Seller)
- **subMerchant / city:** Sub-establishment City (Seller)
- **subMerchant / state:** State of the Subestablishment (Seller)
- **subMerchant / country:** Country of Sub-establishment (Seller)

- **subMerchant / cep:** Cep of Sub-establishment (Seller)
- **subMerchant / mcc:** Branch code/MCC of the sub-establishment (Seller) = Dynamic MCC

To correctly classify the Seller's MCC, the rule determined by ABECs described below must be followed:

MCC Definition Rule:

1. Primary CNAE (National Code of Economic Activity), assigned by the Federal Revenue to classify the establishment's area of activity. ABECs uses a database of CNPJs sent by Serasa, with DE-TO information from CNAE to CNPJ. **To classify the client in any other CNAE, even if it is the secondary CNAE, it is necessary to first demonstrate to the brands the activity carried out by the client.**
2. Assessment in the Brand committee that overrides rule 1. Approval is made in exceptional cases, where the CNAE/MCC rule does not reflect the client's activity. The assessment is carried out by CNPJ and applied after a defense to ABECs.

Note: For more details and access to the base, see the official ABECs website:

<https://www.abecs.org.br/consulta-mcc-individual>

- **subMerchant / CPF_cnpj:** CNPJ/CPF of sub-establishment (Seller)

Request

```
{
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "235",
  "softDescriptor": "LOJADOZE",
  "PaymentFacilitatorID": 22349202212,
  "IndependentSalesOrganizationID": 1234567,
  "subMerchant": {
    "mcc": "1111",
    "subMerchantID": 1234567890,
    "address": "Rua Acre",
    "city": "CAPIVARI",
    "state": "SP",
    "country": "BRA",
    "zipcode": "07064-010",
```

```

"tin": "71.789.371/0001-42",
"taxIdNumber": "999999999999"
}
}

```

Requisition parameters:

Name	Size	Type	Mandatory	Description
softDescriptor	Up to 18*	Alphanumeric	Yes*	Personalized phrase that will be printed on the cardholder's invoice.
paymentFacilitatorID	Up to 11	Numeric	Yes*	Facilitator code.
independentSalesOrganizationID	Up to 11	Numeric	No	Independent sales organization code.
subMerchant		subMerchant		
subMerchant / mcc	4	Alphanumeric	Yes*	MCC of the sub shopkeeper.
subMerchant / subMerchantID	Up to 15	Alphanumeric	Yes*	Sub shopkeeper's Code
subMerchant / address	Up to 48	Alphanumeric ¹	No*	Sub shopkeeper's Address
subMerchant / city	Up to 13	Alphanumeric ¹	No*	Sub shopkeeper's City
subMerchant / state	2	Alphabetical	Yes*	Sub shopkeeper's State
subMerchant / country	Up to 3	Alphanumeric	Yes*	Sub shopkeeper's Country
subMerchant / cep	Up to 9	Alphanumeric	Yes*	Sub shopkeeper's Postal code
subMerchant / cnpj	Up to 18	Numeric	No*	Sub shopkeeper's CNPJ (TIN)
subMerchant / taxIdNumber	Up to 14	Alphanumeric	Yes*	Sub shopkeeper's CNPJ (TIN) or CPF (ITIN)

Note that request parameters inside the “Submerchant” group must always start with a lowercase letter.

Important: Due to the LGPD (General Data Protection Law), the following fields of the “SubMerchant” key: SubMerchantID, Address, City, State, Country, Zip Code and CNPJ (TIN), even when sent in the request, are not returned in transaction queries.

Attention:

- To ensure proper processing of the transaction, special characters must not be included.

Response

```
{
  "reference": "160224094727",
  "tid": "10012402160947499092",
  "nsu": "557854383",
  "dateTime": "2024-02-16T09:47:49-03:00",
  "amount": 1000,
  "cardBin": "544828",
  "last4": "0007",
  "brand": {
    "name": "Mastercard",
    "returnMessage": "Success.",
    "returnCode": "00",
    "brandTid": "263942",
    "authorizationCode": "263942"
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/10012402160947499092"
    },
    {
      "method": "POST",
      "rel": "refund",
      "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/10012402160947499092/refunds"
    }
  ]
}
```

Response parameters:

Name	Size	Type	Description
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh: mm: ss.STZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.

Name	Size	Type	Description
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones. For more details see the section Recurrence and Card-on-file
returnCode	Up to 3	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.

Digital wallets

Wallets on e.Redre work as devices that store cards and payment data for e-commerce buyers. They allow the consumer to register their payment credentials and be able to make payments quickly and conveniently by cell phone or other connected devices, for example.

The Wallets that e.Redre can receive transactions are:

- [Apple Pay](#)
- [Google Pay](#)
- [Samsung Pay](#)

By clicking on each of the links above, you can access the official website of each of the Wallets with information for integration of check-out and transactional flow.

At this time, e.Redre only processes these transactions, that is, the establishment must be or have an intermediary (such as a gateway) that is a PSP (Payment Service Provider). PSPs have integration with Wallets to decrypt transaction data and forward it to the acquirer (Redre) for processing.

Soon, the Redre Payments Platform will also offer the PSP solution in order to facilitate the integration of our customers.

At the moment, the Payment Platform Redre processes transactions from Wallets (Apple pay, Google Pay, Samsung Pay) under **Visa and Mastercard** brands.

In addition to these options, the Mastercard, Visa, Elo and Amex brands have a Stored Digital Wallet program, check [here](#)

POST /v1/transactions

Request

```
{
  "capture": "true",
  "kind": "credit",
  "reference": "ss112",
  "amount": "100",
  "cardNumber": "2223000250000004",
  "expirationMonth": "2",
  "expirationYear": "2030",
  "securityCode": "268",
  "storageCard": "2",
  "wallet": {
    "processingType": "04",
    "walletCode": "GEP"
  },
  "securityAuthentication": {
    "sai": "05"
  },
  "transactionCredentials": {
    "credentialId": "01"
  }
}
```

Requisition parameters:

Name	Size	Type	Mandatory	Description
capture		Boolean	No	<p>Define whether a transaction will be captured automatically or later. Failure to submit this field will be considered an automatic capture (true).</p> <p>For debit and Zero Dollar transactions, when this field is sent, the parameter must be set as true, indicating automatic capture.</p>

Name	Size	Type	Mandatory	Description
kind		Alphanumeric	No	Type of transaction to be performed. <ul style="list-style-type: none"> For credit transactions, use credit For debit transactions, use debit <p>Failure to submit this field will be considered credit.</p>
reference	Up to 16	Alphanumeric	Yes	Order code generated by the establishment.
amount	Up to 10	Numeric	Yes	Total transaction amount without thousands and decimal separators. Examples: <ul style="list-style-type: none"> R\$10.00 = 1000 R\$0.50 = 50
installments	Up to 2	Numeric	No	Number of installments in which a transaction will be authorized. From 2 to 12 Failure to submit this field will be considered in cash.
cardholderName	Up to 30	Alphanumeric	No	Cardholder's name printed on the card. Do not send special characters
cardNumber	Up to 19	Alphanumeric	Yes	Card number.
expirationMonth	Up to 2	Numeric	Yes	Card expiration month. From 1 to 12.
expirationYear	2 or 4	Numeric	Yes	Card expiration year E.g.: 2028 or 28
securityCode	Up to 4	Alphanumeric	No	The card security code is usually located on the back of the card. Sending this parameter guarantees a greater possibility of approval of the transaction.

Name	Size	Type	Mandatory	Description
storagecard	Up to 18 *		No	Indicates operations that may or may not be using COF (Card on File): 0 - Transaction with credential not stored. 1 - Transaction with credential stored for the first time. 2 - Transaction with credential already stored. Attention: Failure to send this field will be considered 0 (credential not stored).
tokenCryptogram		Alphanumeric	Mandatory (See more details at the end of the table)	Token informed by the Card Brand. Identifies tokenized transactions.
wallet				Wallet group for walletId and walletCode parameters
wallet/processing type	2	Alphanumeric	Yes	Operation type identification for Apple, Google and Samsung Pay: • Use 03 for ELO brand; • Use 04 for Visa and Mastercard brands To check the SDWO information see the section Staged Digital Wallet Operators (SDWO)
wallet/walletId	Up 11	Numeric	Mandatory for ELO transactions (processingType=3)	Identifies the Wallet originating the transaction, they are fixed and mandatory IDs for Elo use. 52810030273 – Apple Pay 52894351835 – Google Pay 52815860843 – Samsung Pay
wallet/walletCode	Up 3	Alphanumeric	Yes	Identifies the Wallet, exclusive use for processingType=03 or 04 AEP = Apple Pay GEP = Google Pay SGP = Samsung Pay

Name	Size	Type	Mandatory	Description
securityAuthentication	-	-	-	securityAuthentication group
securityAuthentication /sai	Up 2	Alphanumeric	Yes for Visa and ELO brands.	<p>Electronic Transaction Identifier (ECI).. In transactions that are not tokenized (only card-on-file), sending this field is not necessary.</p> <p>For Wallets transactions with Elo card, always send the value “04” – indicating an in-app transaction and for Wallets with Visa and Mastercard card, follow what was sent by Wallet.</p> <p>For Mastercard branded transactions, this field is not sent</p> <p>For more details on this field, check the topic “using sai”.</p>
transactionCredentials	-	-	-	transactionCredentials Group
transactionCredentials/credentialId	Up to	Alphanumeric	Yes, if storagecard=1 or storagecard=2 and mastercard brand	Indicates the category of transaction with stored credential. See the “Categorizing Card-on-File Transactions” section for more details.

We emphasize that Wallets use brand tokenization in their transactions. In this way, when the buyer saves the card, a brand token is created, thus changing the information on the physical card (number, security code and validity).

Therefore, it is mandatory to send the card number + cryptogram token fields returned by each wallet in transactions. Furthermore, in MIT transactions (Initiated by the establishment) for the Visa brand, it is allowed to send the tokenized card number (cardnumber field), without the tokencryptogram field, maintaining the value of the “sai” field indicated by Wallet. For other brands, the Cryptogram token must be sent.

Use of “sai”: The parameter must be used whenever the transaction has a specific ECI, which is not linked to 3DS authentication (for example: Wallets and authentication of brand tokens), when authenticated through a 3DS challenge it is necessary that “eci” is informed within the 3D Secure group, and it is not necessary to use “sai” in this case. **For Wallets transactions with an Elo card, always send the value “04” – indicating an in-app transaction** and for Wallets with a Visa card, follow what was sent by the Wallet.

Attention points:

- When sending the threeD Secure group in any request, the “sai” field will be ignored and the “eci” from the threeD Secure group will be prioritized. If the parameters are sent incorrectly, the brand may downgrade the security level of the transaction, that is, it may classify it as unauthenticated, losing the issuing liability. Pay attention to the parameters requested in the documentation.
- Wallets transactions can also receive a chargeback dispute if they have an ECI (sent in the “sai” field) for an unsecured transaction. Observe the parameters sent by Wallets and forward them in your e.Redem requests to ensure that brands and issuers receive the information in its entirety.

Use of “sai”: The parameter must be used whenever the transaction has a specific ECI, which is not linked to 3DS authentication (ex: Wallets and Cloud Token Visa), when authenticated as 3DS it is necessary that the “eci” be informed within the 3D Secure group, it is not necessary to use the “sai” in this case.

Response:

```
{
  "reference": "445295211",
  "tid": "10032204221000000007",
  "nsu": "881709007",
  "dateTime": "2022-04-22T14:55:36-03:00",
  "amount": 200,
  "installments": 2,
  "cardBin": "491002",
  "last4": "8004",
  "brand": {
    "name": "Mastercard",
    "returnCode": "00",
    "returnMessage": "Success.",
    "authorizationCode": "F8RNNN",
    "brandTid": "F8RNNN"
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
```

```
{
  "method": "GET",
  "rel": "transaction",
  "href": "https://api.userede.com.br/eredede/v1/transactions/10032204221444268457"
},
{
  "method": "POST",
  "rel": "refund",
  "href":
  "https://api.userede.com.br/eredede/v1/transactions/10032204221444268457/refunds"
},
{
  "method": "PUT",
  "rel": "capture",
  "href": "https://api.userede.com.br/eredede/v1/transactions/10032204221444268457"
}
]
}
```

Response parameters:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh:mm:ss.TZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.

Name	Size	Type	Description
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones. For more details see the section Recurrence and Card-on-file

Attention: Visa cards that are tokenized by Wallets after July 30th, 2025 will not be able to carry out installment or recurring transactions. In these cases, cards must be tokenized via card-on-file, with tokens for use by the establishment.

Apple Pay

Apple Pay is Apple's digital wallet available on Apple devices such as:

- iPhone (models with Touch ID, Face ID, except 5s),
- Apple Watch (Apple Watch Series 1 and subsequent),
- Mac (Models with Touch ID)
- iPad (iPad Pro, iPad Air, iPad and iPad mini with Touch ID or Face ID).

Payment via Apple Pay replaces the card data with a token, so it works as a tokenized transaction, making the transaction more secure.

In order to offer Apple Pay to your customers, you need to affiliate with Apple and Apple Pay or have an integrated partner such as a PSP capable of decrypting the Wallet payload and then sending it following the integration instructions with e.Rede. You can find the detailed information by searching for "Apple Pay" in the technology documentations of the [Apple Developer Portal](#). In addition, it is essential that your buyers are accessing the website through the Safari browser or through the App on an Apple Pay compatible iOS device.

Google Pay

Google Pay is Google's virtual wallet, available on many Android devices. It allows consumers to make payments, conveniently and securely, with their stored credit and debit cards.

Payment via Google Pay replaces the card data with a token, so it works like a tokenized transaction, making the transaction more secure.

To perform the integration it is necessary that your establishment has the registration and integration with Google Pay or have an integrated partner such as PSP capable of decrypting the Wallet payload and then sending it following the instructions for integration with e.Rede

For Google Pay, there are two types of credentials:

- **Tokenized:** Cards saved through the Issuer's Google Wallet app, promote liability shift and are stored in the wallet.
- **Conventional:** Cards originating from the onboarding process through Google Autofill or Google Settings/Account (pay.google.com). These cards are Card on File and Google Pay recognizes the device and presents it with each and every purchase - they do not promote liability shift. In this case, it is recommended to use 3DS authentication or other transaction security mechanisms as anti-fraud.

For both, there are parameters in the Google API that allow you to identify whether the transaction comes from a tokenized credential or not - this string is called AssuranceDetails in the Google Pay documentation.

For more details of the Google Pay integration above check the [Google Developer Portal](#).

Samsung Pay

Samsung Pay is Samsung's digital wallet, available on newer devices, lets you load your credit, debit, gift, and membership cards onto your devices. With it you can make payments and authenticate your purchase with your fingerprint, PIN, or iris scan.

Payment through Samsung Pay replaces the card data with a token, which is a unique random set of numbers to be used in each new transaction, so that the actual card number is never used, making the transaction more secure.

To carry out the integration, your establishment must register and integrate with Samsung Pay or have an integrated partner such as a PSP capable of decrypting the Wallet payload and then sending it following the integration instructions with e.Red. For more details of the integration check the [Samsung Pay](#) website.

Returns from Wallets transactions follow those of common transactions on e.Red, available on our developer portal. It is important to be aware of all possible [Integration Returns](#).

Also be aware of the [Returns provided by the brand](#), also available on our developer manual and which may indicate negatives on their part.

Staged Digital Wallet Operators (SDWO)

A digital wallet is an electronic solution that allows financial and identity data to be stored in a way that enables them to be used securely and privately during financial transactions.

There are two types of digital wallets- *Staged Digital Wallets* -

- Cash-in
 - The wallet is supplied with funds through a financial transaction using the card previously registered on its platform, for later use;

Attention: Cash-in transactions for Elo brand cards are only allowed for debit cards, pre-paid cards and credit cards without installments
- Purchase
 - The wallet performs a financial transaction to a partner merchant or transfers values between wallets, using the card previously registered on its platform.

For the correct functioning of the integration of the digital wallet service, besides the initial integration of the transactional flow ([Authorization](#)), it is necessary that some other integrations to our API's have already been performed:

- [Dynamic MCC \(Merchant Category Code\)](#)
- [softDescriptor](#)

The SDWO (Staged Digital Wallet Operators) service, can be used in conjunction with the other services available on the Rede:

The Consumer Bill Payment Service (CBPS)*

(*) CBPS + Digital Wallet transactions are possible in Visa brand. From April, 2024 will be available for Amex and Mastercard.

The digital wallet service can be used on the Elo, Mastercard, Visa brands and from April 2024, on the Amex brand.

For the correct identification of a transaction of the Digital Wallets type, there are some fields that need to be filled in the transactional flow according to the modality, according to the rules specified by card brands. Check below the list of existing fields and their respective formats in Staged Digital Wallet operations. The expected rules for each brand will be presented in the sequence:

Requisition parameters:

Name	Size	Type	Mandatory	Description
capture		Boolean	No	<p>Define whether a transaction will be captured automatically or later. Failure to submit this field will be considered an automatic capture (true).</p> <p>For debit and Zero Dollar transactions, when this field is sent, the parameter must be set as true, indicating automatic capture.</p>
kind		Alphanumeric	No	<p>Type of transaction to be performed.</p> <ul style="list-style-type: none"> • For credit transactions, use credit • For debit transactions, use debit <p>Failure to submit this field will be considered credit.</p>

Name	Size	Type	Mandatory	Description
reference	Up to 16	Alphanumeric	Yes	Order code generated by the establishment.
amount	Up to 10	Numeric	Yes	Total transaction amount without thousands and decimal separators. Examples: <ul style="list-style-type: none"> R\$10.00 = 1000 R\$0.50 = 50
cardNumber	Up to 19	Alphanumeric	Yes	Card number.
expirationMonth	Up to 2	Numeric	Yes	Card expiration month. From 1 to 12.
expirationYear	2 or 4	Numeric	Yes	Card expiration year E.g.: 2028 or 28
securityCode	Up to 4	Alphanumeric	No	The card security code is usually located on the back of the card. Sending this parameter guarantees a greater possibility of approval of the transaction.
softDescriptor	Up to 18*	Alphanumeric	Yes*	Personalized phrase that will be printed on the cardholder's invoice. Check the standard to be followed in each operation with more details in the specifications below

Name	Size	Type	Mandatory	Description
subMerchant				Group subMerchant
subMerchant / mcc	4	Numeric	Yes*	MCC of the sub shopkeeper.
wallet				Group of Wallet Data
wallet/walletId	Up to 11	Alphanumeric	See rules by brand	which is the identification number of the wallets with each one of the card brands.
wallet/processingType	2	Numeric	Yes	Identifies the type of operation if it is a Cash-in (02) or Purchase (01).
wallet/paymentDestination	2	Numeric	See rules by brand	Identifies the destination/purpose of the cash-in: <ul style="list-style-type: none"> • 04: M2M (Same ownership, same portfolio/arrangement) • 05: P2P (For another holder, same wallet/arrangement) • 06: Transfer to another arrangement (same ownership) • 07: Transfer to another arrangement (other ownership)
receiverData				Group of receiver Data
receiverData/firstName	Up to 40	Alphanumeric	See rules by brand	First name of the cash-in recipient. Do not use special characters.
receiverData/lastName	Up to 40	Alphanumeric	See rules by brand	Last name of the cash-in recipient. Do not use special characters.
receiverData/taxIdNumber	Up to 14	Numeric	See rules by brand	CPF or CNPJ of the cash-in recipient.

Name	Size	Type	Mandatory	Description
receiverData/ walletAccountIdentification	Up to 50	Numeric	See rules by brand	User identifier in the wallet
consumerBillPaymentService				Group for Bill Payment Service
consumerBillPaymentService / businessApplicationIdentifier	2	Numeric	Yes, if bill payment operation	CBPS transaction identifier. For this type of transaction, this field must be filled in with "01". When we do not have this type of transaction, the field should not be sent
consumerBillPaymentService / merchantTaxId	Up to 14	Numeric	No	Identifier of the final beneficiary/transferor of the bill. CPF/CNPJ must be informed. For Mastercard, if it is not indicated, it will be considered an unidentified bill.

CASH-IN:

POST /v1/transactions

Request for Cash in + Digital Wallets

```
{
  "capture": true,
  "reference": "pedido456",
  "amount": 2000,
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "tokenCryptogram": "ANbuvvxndbK2AAEShHMMWgADFA==",
  "securityCode": "123",
  "softDescriptor": "LOJADOZE",
  "subMerchant": {
    "mcc": "6540",
  },
  "wallet": {
```

```

    "walletId": "3900370000",
    "processingType": "02",
    "paymentDestination": "04",
    "receiverData":{
      "firstName": "Jose",
      "lastName": "Silva",
      "taxIdNumber": "31412311177
      "walletAccountIdentification": "342432409",
    }
  },
  "securityAuthentication": {
    "sai": "01"
  }
}

```

Requisition Parameters:

See below the list of mandatory fields for the operation according to each brand.

	Elo	Mastercard	Visa	Amex
softDescriptor	Name of the Wallet * Name of the receiving account holder in the Wallet	Name of the Wallet * Name of the receiving account holder in the Wallet	Name of the Wallet * Name of the receiving account holder in the Wallet	Name of the Wallet * Name of the receiving account holder in the Wallet
submerchant				Submerchant group
submerchant/mcc	6051	6540	6051	6540
wallet				Wallet group
wallet/processingType	02	02	02	02
wallet/walletId	11 characters (See how to obtain at the end of the tables)	3 characters (See how to obtain at the end of the tables)	10 characters, if the parameter has less than 10 characters you must fill in the number of zeros missing to the right.	Send the first 8 digits of the wallet's CNPJ

			Ex: 3900370000 (See how to obtain at the end of the tables)	
wallet/paymentDestination	-	Identifies the destination/purpose of the cash-in: <ul style="list-style-type: none"> • 04: Me2Me (Same ownership, same portfolio/arrangement) • 05: P2P (For another holder, same wallet/arrangement) • 06: Transfer to another arrangement (same ownership) • 07: Transfer to another arrangement (other ownership) 	-	-
receiverData/ Group of receiver Data				
receiverdata/firstName	-	First name of the cash-in recipient. Do not use special characters.	-	-
receiverdata/lastName	-	Last name of the cash-in recipient. Do not use special characters.	-	-
receiverdata/taxIdNumber	CPF or CNPJ of the cash-in recipient.	CPF or CNPJ of the cash-in recipient.	CPF or CNPJ of the cash-in recipient.	-
receiverdata/walletAccountIdentification	-	User identifier in the wallet	-	-

PURCHASE:

POST /v1/transactions

Request for Purchase + Digital Wallets

```
{
  "capture": true,
  "reference": "request789",
  "amount": 2099,
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "123",
  "tokenCryptogram": "ANbuvvxndbK2AAEShHMWgGADFA==",
  "softDescriptor": "string",
  "subMerchant": {
    "mcc": 1111,
  },
  "wallet": {
    "walletId": 1234567890,
    "processingType": "02"
  }
}
```

Requisition Parameters

See below the list of mandatory fields for the operation according to each brand.

	Elo	Mastercard	Visa	Amex
softDescriptor	Name of the Wallet * Name of the receiving merchant	Name of the Wallet * Name of the receiving merchant	Name of the Wallet * Name of the receiving merchant	Name of the Wallet * Name of the receiving merchant
submerchant				Submerchant group
submerchant/mcc	MCC of merchant	MCC of merchant	MCC of Merchant, except 6010 and 6011	MCC of merchant
wallet				Wallet Group
wallet/processingType	01	01	01	01

wallet/walletId	11 characters (See how to obtain at the end of the tables)	3 characters (See how to obtain at the end of the tables)	10 characters, if the parameter has less than 10 characters you must fill in the number of zeros missing to the right. Ex: 3900370000 (See how to obtain at the end of the tables)	Send the first 8 digits of the wallet's CNPJ
------------------------	---	--	--	--

CBPS:

POST /v1/transactions

Request for CBPS + Digital Wallets

```
{
  "capture": false,
  "reference": "c202012291716",
  "amount": 2099,
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2028,
  "securityCode": "123",
  "softdescriptor": "string",
  "subMerchant": {
    "mcc": 1111,
  },
  "consumerBillPaymentService": {
    "businessApplicationIdentifier": "01"
    "merchantTaxId": "11122233344"
  },
  "wallet": {
    "walletId": 1234567890,
    "processingType": "02",
  }
}
```

Requisition Parameters

See below the list of mandatory fields for the operation according to each brand.

	Mastercard	Visa	Amex
softDescriptor	Name of the Wallet * Name of the receiving merchant	Name of the Wallet * Name of the receiving merchant	Name of the Wallet * Name of the receiving merchant
submerchant			Submerchant group
submerchant/mcc	Check the list of permitted MCCs here	Check the list of permitted MCCs here	Check the list of permitted MCCs here
consumerBillPaymentService			Group for Bill Payment Service
consumerBillPaymentService/businessApplicationIdentifier	01	01	01
consumerBillPaymentService/merchantTaxId	CNPJ of the invoice/final beneficiary. If not sent, it is considered as an unidentified bill	-	CNPJ of the invoice/final beneficiary.
wallet			Wallet Group
wallet/processingType	02	02	02
wallet/walletId	3 characters (See how to obtain at the end of the tables)	10 characters, if the parameter has less than 10 characters you must fill in the number of zeros missing to the right. Ex: 3900370000 (See how to	Send the first 8 digits of the wallet's CNPJ

		obtain at the end of the tables)	
--	--	-------------------------------------	--

Response for all Operations of SDWO

```
{
  "reference": "090224151745",
  "tid": "10402402090947380022",
  "nsu": "2330041",
  "dateTime": "2024-02-09T15:17:47-03:00",
  "amount": 13400,
  "cardBin": "506723",
  "last4": "9011",
  "brand": {
    "name": "Elo",
    "returnCode": "00",
    "returnMessage": "Success.",
    "authorizationCode": "524249",
    "brandTid": " 012345678912345000010"
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://api-hom.userede.com.br/eredev1/transactions/10402402090947380022"
    },
    {
      "method": "POST",
      "rel": "refund",
      "href": "https://api-hom.userede.com.br/eredev1/transactions/10402402090947380022/refunds"
    }
  ]
}
```

Response for Operations of SDWO:

Name	Size	Type	Description
returnCode	Up to 4	Alphanumeric	Transaction return code.

Name	Size	Type	Description
returnMessage	Up to 256	Alphanumeric	Transaction return message.
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh:mm:ss.TZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones. For more details see the section Recurrence and Card-on-file

Pay attention to our [integration returns table](#)

- Attention: The request for generating the Wallet ID for the Master and Visa card brands is made by the Rede facilitators team (facilitadores@userede.com.br). For the Elo card brand, the registration must be done directly by the customer via e-mail to aceutacaofacilitadores@elo.com.br.
- For the Amex brand: walletId is not registered now, so the first 8 CNPJ numbers must be sent in the walletId field to identify the establishment. In the future, if the brand creates specific walletIds, customers will be notified.

Use of “sai”: The parameter must be used whenever the transaction has a specific ECI, which is not linked to 3DS authentication (ex: Wallets and Cloud Token Visa), when authenticated as 3DS it is necessary that the “eci” is informed within the 3D Secure group, it is not necessary to use the “sai” in this case.

Attention: When sending the threeD Secure group in any request, the “sai” field will be ignored and the 3DS stream will be prioritized.

Pix

Definition of Pix

It is a payment, receipt and transfer option within digital wallet and bank applications - between individuals or legal entities.

Attention: Payment method available only for Itaú account holders.

Pix key registration

To enable your Itaú key on the Rede Network:

1. Go to userede.com.br;
2. Login in your Rede account;
3. Go to rote Para vender > PIX > “Quero utilizar Pix” > accept our user terms > Select your agency and current account.

Be careful not to delete your key before completing operations in your transactions, such as requesting returns of paid QR Codes.

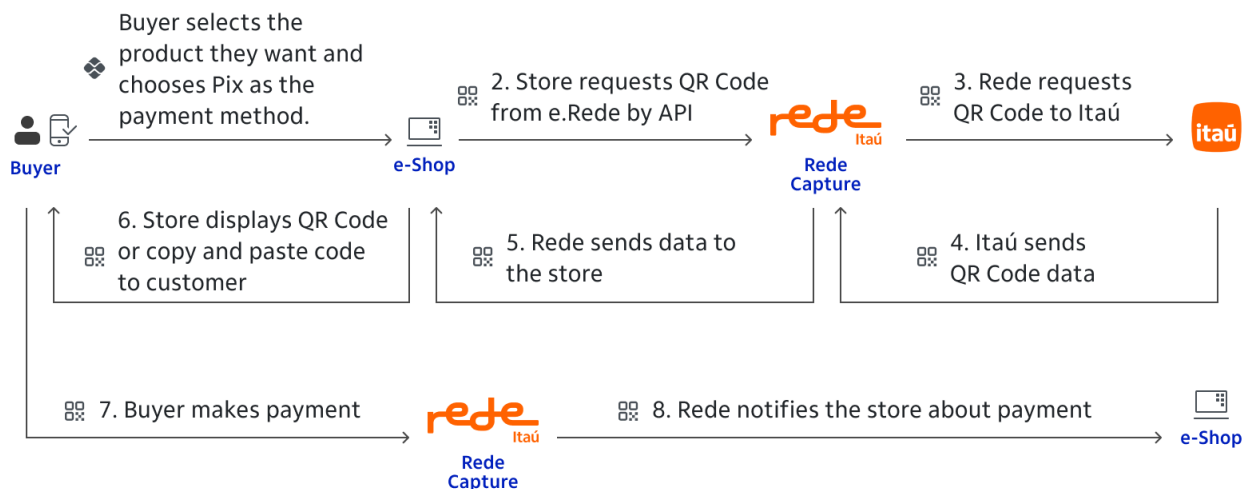
For pricing questions, consult the call center or the value negotiated in your Itaú account.

QR Code Pix request

Transaction flow

To detail the transactional flow, see below the request, payment, and receipt of payment notifications flow.

Pix on e.Redre - Transactional Journey



QR Code request example:

POST /v1/transactions

```
{
  "kind": "pix",
  "reference": "pix{{reference}}",
  "amount": "3900",
  "qrCode": {
    "Date timeExpiration": "2023-09-30T13:15:59"
  }
}
```

Description of fields:

Name	Size	Type	Mandatory	Description
kind		Alphanumeric	Yes	Type of transaction to be carried out. For pix transactions, use pix
reference	Until 16	Alphanumeric	Yes	Order code generated by the establishment.
amount	Until 10	Numeric	Yes	Total transaction amount without thousands and decimal separator. Examples: R\$10.00 = 1000
qrCode				Group qrCode
qrCode/Date timeExpiration		Date and time	Yes	QRcode expiration data in YYYY-MM-DDThh:mm:ss format. The maximum period must be up to 15 days and must not be from an

earlier date than the current one.

QR Code response example:

```
{
  "reference": "pix310723140848",
  "tid": "40402307310827210012",
  "Date time": "2023-07-31T14:08:50-03:00",
  "amount": 3900,
  "qrCodeResponse": {
    "Date timeExpiration": "2023-08-30T16:48:00-03:00",
    "qrCodeImage":
      "iVBORw0KGgoAAAANSUHEUgAAAPoAAAD6AQMAAACyIsh+AAAAB1BMVEX///8AAABVwtN+AAAC/E1EQVR42uy
      ZPY6t0BCFj+XAoXeAN4JgWY9AMhLB2xaoN2J2UKEDxB1V0X2ZpzfhNNdBV0TjryVj18+puvixH3vcEkmZkuS
      N1ZeaCzIlbCTZEABAc0lcEdYUZsd5l0rKtdIKEFhk6uLvjTwAdHGhAjyaA5IAMazgbEfP7XkAQF6jXvFskDc
      2B9h175LPiB6Bu0yj4C9/eDNggapXLbVPYdl1GuLfkfwTwmTg6iYPxGUj1/9INW8EUvUUfyKcypC7f1VVL2B
      DQDDn0wxTe1RNiGvU1dwQgPBh0R1WY/Iuk517fZ31A4BaLnEeIjyBJLo6XimvGYBH0o1nClnC4QR0gj6lDoB
      UxxKXEr1RczVPO/EhwZfngHCAB+JvXUmCHZeN1V43BNSRQAc4u2f1ScfqSpGbAlSYkCVWxHBohUPetPqHpTQ
      E0LKJqDMoemICAqmy7ykg1T5JLghr1D+mDv6M4QTvo34/APSIPgSaKvYnvFU0LmwIsJSXOA966Jr8tMLpUT8
      HJIyvjuFAsAxj0vPwCg0ApooR51EsLih+BbdS0RJQvX4F5y8dZT653tfdBqDqyRoxfZr1exjUNcpjAMwNNS6
      0sGLqkE9UV26nbQEIH4WH0+1Xz6BSwELoD6H1fmBNmBCrU3lQPbX41uFfIuv7gWTvEOuohfXrrSt/SPd3A3q
      ScXbk1WHmznrXFZhs00BCD/H71R2misRF02B6Vf8mAJW/Wnm1jJSqIt7JVWQeA4CxCFRpinpnLpGn/d/cFNA
      nTJ2NvGj1zZ9aL+5msAnAF8kn8qreaUOSAdps3J3a9w08tvZrpIaxFdaIwVygGUALijLmk2bHYM0v0bUDXG3
      1qd2iVv/Z8XBiUwC2A9hMTAPnU0ft2i2GrdQJzwEAZOrwC/Ha5Ne89U6DDQCf42javJDLzuWS7p6NAX6XsEn
      tQe7iz6i5enkYs0tm+KD6Wz41Pd/X3QLw+QuIAGeQd4FjdDoZNGTYvFrjF2HFa8gEeLYD/NiP/Y/2TwaAAP/
      /gw0NUkoPVIYAAAAASUVORK5CYII",
    "qrCodeData": "00020101021226720014br.gov.bcb.pix2550bx.com.br/pix/8b3da2f3-
      9a41-40d1-a91a-bd93113bd4415204000053039865406123.455802BR5913Fulano de
      Tal6008BRASILIA62190515RP12345678-2019630445C8"
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://api-
      hom.userede.com.br/eredev1/transactions/40402307310827210012"
    },
    {
      "method": "POST",
      "rel": "refund",
      "href": "https://api-
      hom.userede.com.br/eredev1/transactions/40402307310827210012/refunds"
    }
  ]
}
```

Description of fields:

Name	Size	Type	Description
reference	Until 16	Alphanumeric	Order number generated by the merchant

Name	Size	Type	Description
tid	Until 20	Alphanumeric	Unique transaction identifier number..
Date time		Date and time	QrCode request date in the format YYYY-MM-DDhh:mm:ss.sTZD .
amount	Until 10	Numeric	Total transaction amount without thousands and decimal separators
qrCodeResponse			Grupo QrCode
qrCodeResponse/Date timeExpiration		Date and time	Date expiration and QrCode of the transaction in the formatto YYYY-MM-DDhh:mm:ss.sTZD .
qrCodeResponse/qrCodeImage	Until 999	Alphanumeric	Field with QRCode string in base64. To render the qrCode image, simply use libraries or scripts compatible with your programming language.
qrCodeResponse/qrCodeData	Until 999	Alphanumeric	field with string do QRCode in format emv (copy and paste)
returnCode	Until 04	Alphanumeric	Request return code qrCode.
returnMessage	Until 256	Alphanumeric	Request return message qrCode.

Status notification by webhook

After requesting a QR Code Pix, each update regarding the payment or return status (**by Itaú channels**), a notification will be returned at the url provided by the merchant.

To register, the establishment must call the call center on the following telephone numbers: Call center: Capitals and metropolitan regions 4001 4433 or Call center: Other locations 0800 728 4433 and inform the CNPJ number, PV, contact email and URL you want to use to receive notifications from Pix. The deadline for activation is 2 business days after opening the call. The registration of this URL will be by CNPJ, regardless of how many or which PVs were enabled for that establishment.

To request the registration of your URL, contact us through our service channels and inform:

- **CNPJ;**
- PV;
- E-mail;
- URL that will be used to receive Rede payment notifications.

Possible webhook events:

Event notification	Status
PV.UPDATE_TRANSACTION_PIX	Approved

PV.REFUND_PIX	Canceled (partial or fully)
---------------	-----------------------------

For returns requested by API, there are no notifications, as the success or failure response is given synchronously. For total or partial returns made through other Itaú channels such as bankline, you will receive a notification and will be able to view the list of partial returns in the e.Redre transaction query and cancellation query APIs.

Upon receipt of the notification, it is optional to return to the e.Redre query API with the TID for more transaction details. We recommend that you wait at least 10 minutes to make inquiries after receiving a notification.

IMPORTANT: If the notification “endpoint/url” is not provided, no event will be notified during the payment or return process of your Pix transactions.

To validate how to simulate in a test environment, see the section [Simulação de notificação de status via webhook](#).

Webhook notification example:

```
{
  "companyNumber": "12341088",
  "events": [
    "PV.UPDATE_TRANSACTION_PIX"
  ],
  "data": {
    "id": "40012309181001158845"
  }
}
```

Description of fields:

Name	Local de envio	Size	Type	Description
authorization	header	Until 3	Alphanumeric	Header for authorization of the request at the URL provided by the establishment – during pilot times we request that it be sent via email if you wish to use authentication to send notifications (optional)
request-ID	header	Until 36	Alphanumeric	Unique request identifier
content-Type	header	-	Alphanumeric	Fixed value defined as 'application/json'
id	body	6	Alphanumeric	Unique transaction identifier (TID)
companyNumber	body	9	Alphanumeric	Establishment membership number (PV)

Name	Local de envio	Size	Type	Description
events	body	-	Lista alfanumérica	Name of events that will be informed to the customer. Example: ["PV.UPDATE_TRANSACTION_PIX"], ["PV.REFUND_PIX"]

Query transaction Pix

For Pix transactions, the query follows the e.Redes standard, and can be carried out using the TID and Reference (order number).

Attention: The qrCodeData and qrCodeImage fields will only be returned in the query if the QR code status is **pending**. For paid or returned qrCodes, these fields **will not be returned**.

In case of expired QR codes, you will receive the code 3036 - QrCode Expired.

GET

/v1/transactions

Pending QR Code query example:

```
{
  "requestDate time": "2023-07-31T14:11:42-03:00",
  "qrCodeResponse": {
    "Date time": "2023-07-31T14:08:50-03:00",
    "returnCode": "00",
    "returnMessage": "Success.",
    "affiliation": 90104480,
    "kind": "Pix",
    "reference": "pix310723140848",
    "amount": 3900,
    "tid": "40402307310827210012",
    "status": "Pending",
    "expirationQrCode": "2023-08-30T16:48:00",
    "qrCodeData": "00020101021226720014br.gov.bcb.pix2550bx.com.br/pix/8b3da2f3-9a41-40d1-a91a-bd93113bd4415204000053039865406123.455802BR5913Fulano de Tal6008BRASILIA62190515RP12345678-2019630445C8",
    "qrCodeImage":
    "iVBORw0KGgoAAAANSUHEUgAAAPoAAAD6AQMAAACyIsh+AAAABlBMVEX///8AAABVwtN+AAAC/E1EQVR42uyZPY6tOBCFj+XAoXeAN4JgWY9AMhLB2xaoN2J2UKEDxB1V0X2ZpzfhNNdBV0TjryVj18+puvixH3vcEkmZkuSN1ZeaCzIlbCTZEABAc0lcEdYUZsd5lOrKtdIKEFHk6uLvjTwAdHGhAjyaA5IAMazgbEFP7XkAQF6jXvFSkDc2B9h175LPiB6Bu0yj4C9/eDNggaPXLbVPYdl1GuLfkfwtwMtG6iYPxGUj1/9INW8EUvUUfyKcypC7f1VVL2BDQDDn0wxTe1RNIgVU1dwQgPBh0R1WY/Iuk517fZ31A4BaLnEeIjyBJLo6XimvGYBH0o1nClnc4QR0gj6l doB UxxKXEr1RczVPO/EhwZfngHCAB+JvXUmCHZeN1V43BNSRQAc4u2f1ScfqSpgbAlSYkCVWxHBohUPetPqHpTQE0LJKJdMoemICAqmy7ykg1T5JLghr1D+mDv6M4QTvo34/APSIPGSaKvYnvFU0LmwIsJSXOA966Jr8tMLpUT8HJIYvjuFAsAxj0vPWcg0ApooR51EsLih+BbdS0RJQvX4F5y8dZT653tfdBqDqyRoxfZr1exjUNcpjAMwNNS60sGLqkE9UV26nbQEIH4WH0+1Xz6BSwELoD6H1fmBNmBCrU3lQPbX41uFfIuv7gWTvEOuohfXrrSt/SPd3A3qScXbk1WHmznrXFZhS00BCD/H71R2misRF02B6Vf8mAJW/Wnm1jJSqIt7JVVQeA4CxCFRpinpnLpGn/d/cFNaNTJ2NvGj1zZ9aL+5msAnAF8kn8qreaUOSAdps3J3a9w08tvZrpIaxFdaIwVygGUALijLmk2bHYM0v0bUDXG3
```

```

1qd2iVv/Z8XBiUwC2A9hMTAPnU0ft2i2GrdQJzwEAZOrwC/Ha5Ne89U6DDQCf42javJDLzuWS7p6NAX6XsEn
tQe7iz6i5enkYs0tm+KD6Wz41Pd/X3QLw+QuIAgeQd4FjddoZNgTYvFrjF2HFa8gEeLYD/NiP/Y/2TwAAAP/
/gw0NUkoPVIYAAAAASUVORK5CYII",
  "origin": 1
},
"links": [
  {
    "method": "GET",
    "rel": "refunds",
    "href": "https://api-
hom.userede.com.br/eredev1/transactions/40402307310827210012/refunds"
  },
  {
    "method": "PUT",
    "rel": "capture",
    "href": "https://api-
hom.userede.com.br/eredev1/transactions/40402307310827210012"
  }
]
}

```

Description of fields:

Name	Size	Type	Description
requestDate time		Date time	Request date in format YYYY-MM-DDThh:mm:ss.sTZD .
qrCodeResponse			Grupo qrCode
qrCodeResponse/ Date time		Date time	Date of creation of QrCode of the transaction in the format YYYY-MM-DDhh: mm: ss.sTZD .
qrCodeResponse/ returnCode	Until 04	Alphanumeric	Request return code qrCode.
qrCodeResponse/ returnMessage	Until 256	Alphanumeric	Request return message qrCode.
qrCodeResponse/ affiliation	Until 9	Numeric	Establishment membership number (PV).
qrCodeResponse/ kind	Until 10	Alphanumeric	Payment method used in the transaction (Pix)
qrCodeResponse/ reference	Until 16	Alphanumeric	Order number generated by the establishment.
qrCodeResponse/ amount	Until 10	Numeric	Total transaction amount without thousands and decimal separators
qrCodeResponse/ tid	Until 20	Alphanumeric	Unique transaction identifier number.
qrCodeResponse/ status		Alphanumeric	Transaction status: <ul style="list-style-type: none"> • Approved • Canceled • Pending

Name	Size	Type	Description
qrCodeResponse/expirationQrCode		Date time	Expiration date of QrCode of the transaction in the format YYYY-MM-DDhh: mm: ss.sTZD .
qrCodeResponse/qrCodeImage	Until 999	Alphanumeric	Field with QRCode string in base64. To render the qrCode image, simply use libraries or scripts compatible with your programming language.
qrCodeResponse/qrCodeData	Until 999	Alphanumeric	field with string of QRCode in format emv (copy and paste)

Approved QR Code query example:

Note: TXID will be displayed in Approved QR Code transactions

```
"requestDate time": "2024-06-06T14:06:11-03:00",
  "authorization": {
    "Date time": "2024-06-15T11:31:59-03:00",
    "returnCode": "00",
    "returnMessage": "Success.",
    "affiliation": 90104480,
    "status": "Approved",
    "reference": "v060624140221998",
    "tid": "40402406060920160076",
    "kind": "Pix",
    "amount": 100000,
    "origin": 1,
    "txId": "RELG000021001332453520230905T200143"
  },
  "capture": {
    "Date time": "2024-06-15T11:31:59-03:00",
    "amount": 100000
  },
  "links": [
```

```

    {
      "method": "GET",
      "rel": "refunds",
      "href": "http://localhost:56513/transactions/40402406060920160076/refund
s"
    },
    {
      "method": "POST",
      "rel": "refund",
      "href": "

```

Description of fields:

Name	Size	Type	Description
requestDate time		Date time	Request date in format YYYY-MM-DDThh:mm:ss.STZD .
authorization/ Date time		Date time	Date of creation of QrCode of the transaction in the format YYYY-MM-DDhh:mm:ss.STZD .
authorization/ returnCode	Until 04	Alphanumeric	Request return code qrCode.
authorization/ returnMessage	Until 256	Alphanumeric	Request return message qrCode.
authorization/ affiliation	Until 9	Numeric	Establishment membership affiliation (PV).
authorization/ status		Alphanumeric	Transaction Status: <ul style="list-style-type: none"> • Approved • Canceled • Pending
authorization/ reference	Until 16	Alphanumeric	Order number generated by the establishment .
authorization/ tid	Until 20	Alphanumeric	Unique transaction identifier number .

Name	Size	Type	Description
authorization/ kind	Until 10	Alphanumeric	Payment method used in the transaction (Pix)
authorization/ amount	Until 10	Numeric	Total transaction amount without thousands and decimal separators .
authorization/ origin	Until 2	Numeric	Identifies the origin of the transaction . e.Rede: 1
authorization/ txid	35	Alphanumeric	Identifies the origin of the transaction
capture/Date time		Date time	Payment date of qrCode in the format YYYY-MM-DDThh:mm:ss.STZD .
capture/amount	Until 10	Numeric	Payment amount of qrCode.
refunds/refundId	36	Alphanumeric	Cancellation request return code generated by the Network Rede .
refunds/ refundDate time		Date time	Return date in format YYYY-MM-DDThh:mm:ss.STZD .
refunds/status	Until 10	Alphanumeric	<ul style="list-style-type: none"> • Done (Return effected) • Denied (Return denied)
refunds/amount	Until 10	Alphanumeric	<ul style="list-style-type: none"> • Return value

Query of Pix transaction with multiple partial refunds:

```
{
  "requestDateTime": "2024-06-10T13:06:10-03:00",
  "authorization": {
    "dateTime": "2024-06-06T11:31:59-03:00",
    "returnCode": "00",
    "returnMessage": "Success.",
    "affiliation": 90104480,
    "status": "Approved",
    "reference": "v100624123907089",
    "tid": "40402406100909140013",
    "kind": "Pix",
    "amount": 1000,
    "origin": 1,
    "txId": "RELG000021001332453520230905T200144"
  },
  "capture": {
    "dateTime": "2024-06-06T11:31:59-03:00",
    "amount": 1000
  },
  "refunds": [
    {
      "refundId": "a5ae4808-b8cb-4972-8286-1c164bc04507",
      "refundDateTime": "2024-05-29T11:31:46-03:00",
      "status": "Done",

```

```

    "amount": 100
  },
  {
    "refundId": "dccc61-c342-4796-bce0-b3ff89417dcf",
    "refundDateTime": "2024-05-29T11:31:46-03:00",
    "status": "Done",
    "amount": 100
  },
  {
    "refundId": "19806a7b-697e-4bbd-a875-88c6023e16af",
    "refundDateTime": "2024-05-29T11:31:46-03:00",
    "status": "Done",
    "amount": 100
  }
],
"links": [
  {
    "method": "GET",
    "rel": "refunds",
    "href":
"http://localhost:56513/transactions/40402406100909140013/refunds"
  },
  {
    "method": "POST",
    "rel": "refund",
    "href":
"http://localhost:56513/transactions/40402406100909140013/refunds"
  }
]
}

```

Possible Pix transactions status:

For query form Pix transactions, the following statuses are possible:

Status	Description
Pending	Your QR code has not received payment updates.
Approved	QR Code has received the payment confirmation
Canceled	Transaction canceled

Note: For partially returned transactions, the status will remain as “Approved” Until for the full balance to be returned

For expired transactions, the code will be displayed 3036 – “*QR Code Expired*”

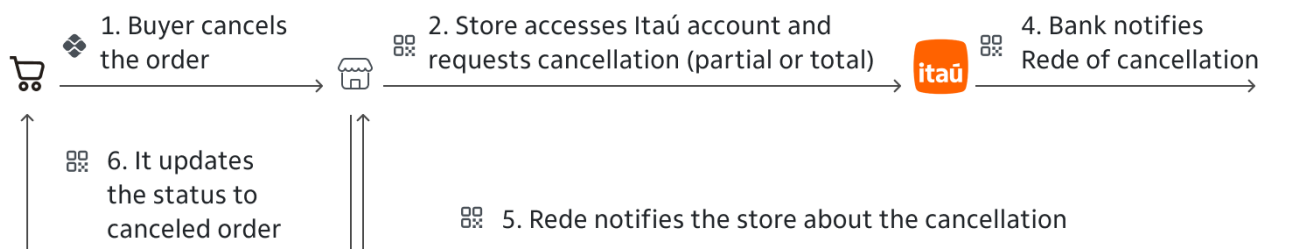
Refund for Pix transactions:

For Pix transactions, refund follows the regulations defined by the Central Bank of Brazil, being allowed up to 90 days from the date of authorization. it will be possible to request a refund **of the total and partial amount from transactions**. The return of request refunds by API will be synchronous, so pay attention to

the return codes that will confirm whether your order was successful, they are available in the section [Return Codes](#).

Note: For refund requests made by Itaú channels, such as bankline, you will receive notifications by webhook in “PV.REFUND_PIX” event (if you have an enabled URL) and will see the list of returns linked to your Pix transaction in the query. For a better understanding of the flow done via bankline, look at the illustration below:

Pix on e.Rede - Cancellation Journey



Refund Request:

POST /v1/transactions/{tid}/refunds

```
{
  "amount": 10000
}
```

Refund Response:

```
{
  "refundDateTime": "2023-09-19T12:12:03-03:00",
  "returnCode": "359",
  "returnMessage": "Refund successful.",
  "links": [
    {
      "method": "GET",
      "rel": "refund",
      "href": "https://sandbox-erede-hom.useredecloud.com.br/v1/transactions/null/refund/null"
    },
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://sandbox-erede-hom.useredecloud.com.br/v1/transactions/null"
    },
    {
      "method": "GET",
      "rel": "refunds",
      "href": "https://sandbox-erede-hom.useredecloud.com.br/v1/transactions/null/refunds"
    }
  ]
}
```

```

    }
  ]
}

```

Description of fields:

Name	Size	Type	Description
refundDate time		Date time	Cancellation date in format YYYY-MM-DDThh:mm:ss.STZD .
returnCode	Until 4	Alphanumeric	Transaction return code (see table return codes for returns).
returnMessage	Until 256	Alphanumeric	Transaction return message (see return codes table for returns).

Return Codes:

Transactions API:

Code	Return message
3076	QrCode: Expiration Date parameter missing.
3077	QrCode: Expiration Date Invalid parameter value.
3078	QrCode: Expiration Date invalid format.
3079	QrCode not processed. Try again.
3081	QrCode: Expiration Date invalid size.
3084	Error generating QrCode Image. Please use the GET Transaction for this operation.
3089	QRCode not generated, please contact Rede
3090	Invalid PIX Key
3092	Fail QrCode generate, please try again;
3094	Unsucessful. Please contact Rede.
3095	Unknown PIX Key.
3096	Unsucessful. Try again later.
3097	Unavailable. Please try again later.
3098	Service not authorized
3099	Comunication failure. Try again later.
51	Product or service disabled for this merchant. Contact Rede
57	affiliation: Invalid merchant
898	PV with invalid ip origin

Query return codes:

Code	Return message
3036	QrCode Expired.

3085	Error generating QR Code Image. Please try again
78	Transaction does not exist

Refunds API:

Code	Return message
3091	Error, not generated. Try again
3094	Unsucessful. Please contact Rede.
3095	Unknown PIX Key.
3096	Unsucessful. Try again later.
3097	Unavailable. Please try again later.
3099	Comunication failure. Try again later.
51	Product or service disabled for this merchant. Contact Rede
57	affiliation: Invalid merchant
78	Transaction does not exist
171	Operation not allowed for this transaction
357	Sum of amount refunds greater than the transaction amount
365	Partial refund not available.
370	Sum of amount refunds greater than the transaction amount
898	PV with invalid ip origin

Sandbox Pix tutorial:

To simulate the QR Code request, queries, returns and return queries, use the instructions in the respective sections presented previously as a basis.

Below, check how to simulate specific error codes and receiving the webhook, which, to support its integration, has some specificities in the testing environment.

Status notification by webhook:

To simulate how webhook notifications are received by Sandbox, send the fields specified below that will allow the registration of Pix events in your Pv Test. In a testing environment, it is not possible to edit or delete the registered URL. If you want to change it, send a new request with the new URL, which will automatically replace the previous one.

Note: In the sandbox environment, payment notification will be made automatically 2 minutes after sending the request to generate a QR Code Pix, simulating a payment (Event "PV.UPDATE_TRANSACTION_PIX").

To simulate receiving webhooks for partial returns made by Itaú channels, send a QR Code request amount R\$50.00. After 2 minutes, you will receive notification in the PV.REFUND_PIX event. Then, if you wish, simulate a consultation to see the cancellation history.

Endpoint for sandbox URL registration: v1/transactions/notification-URL

```
{
  "URL": "https://pixstest.free.beeceptor.com",
  "authorization":{
    "type":"bearer",
    "token": "BEARER 123123"
  }
}
```

Response:

```
{
  "returnCode": "00",
  "returnMessage": "Success"
}
```

Description of fields:

Name	Shipping location	Size	Type	Description
authorization	header	Até 3	Alphanumeric	Header to authorize the request using the URL provided by the establishment (optional)
URL	Body	Até 500	Alphanumeric	Callback URL for sending information to the tokenization requester via the Webhook.
Authorization/type	Body	-	Alphanumeric	Type of authorization to be carried out on the callback URL provided by the establishment. Possible values are: <ul style="list-style-type: none"> • Bearer • Basic
Authorization/token	Body	-	Alphanumeric	Token to be used in the authorization process in the callback URL provided by the establishment. It must be sent in the format 'Bearer XXX' or 'Basic XXX', according to the type provided in the previous field, where the XXX will be the token itself.

Attention: once the authorization object is passed, sending the type and token fields becomes mandatory.

Return Codes:

Code	Return message
S01	Callback URL: Required parameter missing.
S02	Authorization Token: Required parameter missing

S03	Authorization Type: Required parameter missing
S04	Authorization Type: Invalid parameter format
S05	Authorization Token: Invalid parameter format
363	Callback URL characters exceeded 500.
372	Callback URL invalid format (https required).

Simulate return Codes:

To simulate error codes, send requests with the values corresponding to the error codes, as shown in the table below:

Amount	Code	Return message	Ocurs in
123	3036	QrCode Expired.	Query QR Code
3079	3079	QrCode not processed. Try again.	Request QR Code
3081	3081	QrCode: Expiration Date invalid size.	Request QR Code
3084	3084	Error generating QrCode Image. Please use the GET Transaction for this operation.	Request QR Code
3085	3085	Error generating QrCode Image. Please try again	Query QR Code
3089	3089	QRCode not generated, please contact Rede	Request QR Code
3090	3090	Invalid PIX Key	Request QR Code
3091	3091	Error, not generated. Try again	Pix Refund
3092	3092	Fail QrCode generate, please try again;	Request QR Code
3094	3094	Unsucessful. Please contact Rede.	Request/Refund Pix
3095	3095	Unknown PIX Key.	Request/Refund Pix
3096	3096	Unsucessful. Try again later.	Request/Refund Pix
3097	3097	Unavailable. Please try again later.	Request/Refund Pix
3098	3098	Service not authorized	Request/Refund Pix
3099	3099	Comunication failure. Try again later.	Request/Refund Pix
898	898	PV with invalid ip origin	Request/Refund Pix

Card Brands Tokenization (Capture)

The Brand Tokenization service is provided by a Token Requestor, its use can improve the conversion with the brand, because in the case of tokenized cards the card information is protected with due security, replacing it with a token. Each token is unique to the user and establishment and cannot be used by any other store.

When carrying out the transaction, a cryptogram is sent along with the token, preventing card cloning and fraudulent operations. The issuer identifies the use of the token and confirms the authenticity of the cryptogram, thus authorizing the transaction as it knows it is from the genuine bearer.

Currently, the Rede is already prepared to **transact** using Mastercard and Visa tokens. The service that will promote the Tokenization of the brand provided by the Rede is now available in the Visa brand, check it in the [Tokenização de Bandeira Rede](#) menu (not yet available in translated version).

To transact the cryptograms generated by any Token Requestor on e.Redre, check the required fields below and in the API List, as indicated below:

For transactions other than Wallets, sending the tokenCryptogram field is mandatory in all transactions initiated by the bearer (CIT), but optional in transactions initiated by the merchant (MIT).

POST /v1/transactions

Request

```
{
  "capture": true,
  "kind": "credit",
  "reference": "m150420163135479",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "2223000250000004",
  "expirationMonth": 11,
  "expirationYear": 2026,
  "tokenCryptogram": "ANbuvvxndbK2AAEShHMWGgADFA==",
  "storageCard": "1"
  "securityAuthentication": {
    "sai": "01"
  },
  "transactionCredentials": {
    "credentialId": "01"
  }
}
```

Request Parameters

Name	Size	Type	Mandatory	Description
capture		Boolean	No	Defines whether the transaction will be captured automatically or later. Failure to

Name	Size	Type	Mandatory	Description
				send this field will be considered automatic capture (true).
kind		credit / debit	No	Type of transaction to be performed. <ul style="list-style-type: none"> For credit transactions, use credit For debit transactions, use debit Failure to submit this field will be considered a credit.
reference	Up to 16	Alphanumeric	Yes	Order code generated by the establishment.
amount	Up to 10	Numeric	Yes	Transaction value without thousands or decimal separators. <ul style="list-style-type: none"> R\$10.00 = 1000 R\$0.50 = 50
Installments	Up to 2	Numeric	No	Number of installments in which a transaction will be authorized. <p>From 2 to 12</p> Failure to submit this field will be considered in cash.
cardholderName	Up to 30	Alphanumeric	No	Cardholder's name. Do not send special characters
cardNumber	Up to 19	Alphanumeric	Yes	Token number.
expirationMonth	Up to 2	Numeric	Yes	Token expiration month. From 1 to 12.
expirationYear	2 or 4	Numeric	Yes	Token expiration year. Ex: 2028 or 28.
securityCode	Up to 4	Alphanumeric	No	The card security code is usually located on the back of the card. Sending this parameter guarantees a greater possibility of approval of the transaction.

Name	Size	Type	Mandatory	Description
tokencryptogram		Alphanumeric	No	Token informed by the Card Brand. Identify tokenized transactions.
storagecard	1	Alphanumeric	No	<p>Indicates operations that may or may not be using COF (Card on File):</p> <p>0 - Transaction with credential not stored.</p> <p>1 - Transaction with credential stored for the first time.</p> <p>2 - Transaction with credential already stored.</p> <p>For tokenized transactions, this parameter must be sent with a value of "2".</p> <p>Attention: Failure to send this field will be considered 0 (credential not stored).</p>
securityAuthentication	-	-	-	securityAuthentication group
sai	Up to 2	Alphanumeric	Mandatory for Visa and ELO brands. Optional on card-on-file transactions	<p>Electronic Transaction Identifier (ECI). For Mastercard branded transactions, this field is not sent. In transactions that are not tokenized (only card-on-file), sending this field is not necessary.</p> <p>For more details on this field, check the topic "using sai".</p>
transactionCredentials	-	-	-	transactionCredentials Group
transactionCredentials/credentialId	Up to 2	Alphanumeric	Yes, if storagecard=1 or storagecard=2 and	Indicates the category of transaction with stored credential. See the "Categorizing Card-on-File Transactions" section for more details.

Name	Size	Type	Mandatory	Description
			mastercard brand	

Use of “sai”: The parameter must be used whenever the transaction has a specific ECI, which is not linked to 3DS authentication (ex: Wallets and Cloud Token Visa), when authenticated as 3DS it is necessary that the “eci” be informed within the 3D Secure group, it is not necessary to use the “sai” in this case.

Attention: When sending the threeD Secure group in any request, the “sai” field will be ignored and the “eci” from the threeD Secure group will be prioritized.

Sending the securityCode: For Visa, sending the incorrect security code will cause the transaction to be denied.

Response

```
{
  "reference": "160224094727",
  "tid": "10012402160947499092",
  "nsu": "557854383",
  "dateTime": "2024-02-16T09:47:49-03:00",
  "amount": 1000,
  "cardBin": "544828",
  "last4": "0007",
  "brand": {
    "name": "Mastercard",
    "returnMessage": "Success.",
    "returnCode": "00",
    "brandTid": "263942",
    "authorizationCode": "263942"
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://sandbox-
eredede.useredecloud.com.br/v1/transactions/10012402160947499092"
    },
    {
      "method": "POST",
```

```

        "rel": "refund",
        "href": "https://sandbox-
erede.useredecloud.com.br/v1/transactions/10012402160947499092/refunds"
    }
]
}

```

Response parameters:

Name	Size	Type	Description
reference	Up to 16	Alphanumeric	Order number generated by the establishment.
Tid	20	Alphanumeric	Unique transaction identifier number.
Nsu	Up to 12	Alphanumeric	Sequential number returned by the Rede.
authorizationCode	6	Alphanumeric	Transaction authorization number returned by the card issuer.
dateTime		Date and time	Transaction data in the format YYYY-MM-DDhh: mm: ss.sTZD.
amount	Up to 10	Numeric	Total transaction amount without thousands and decimal separators.
cardBin	6	Alphanumeric	6 first digits of the card.
last4	4	Alphanumeric	4 last digits of the card.
brand	-	-	Group of information received from the brand about the transaction
brand/name	-	Alphanumeric	Brand name. Ex: Mastercard
brand/returnCode	Up to 4	Alphanumeric	Transaction return code of brand
brand/returnMessage	Up to 256	Alphanumeric	Transaction return message of brand
brand/merchantAdviceCode	Up to	Alphanumeric	Notice Code for Commercial Establishment. It is a set of codes used to provide additional

Name	Size	Type	Description
			information about a Mastercard exclusive use transaction response.
brand/authorizationCode	6	Alphanumeric	Identifier that differentiates the first recurrence from the subsequent ones.
brand/brandTid	Up to 16	Alphanumeric	Identifier that correlates the first transaction from the subsequent ones. For more details see the section Recurrence and Card-on-file
returnCode	Up to 3	Alphanumeric	Transaction return code.
returnMessage	Up to 256	Alphanumeric	Transaction return message.

Rede Tokenization of Card Brand

The Card brand Tokenization service protects card information by replacing it with a token.

Each token is unique to the user and establishment and cannot be used by any other store.

Once generated, the Establishment can carry out their transactions securely with this token.

Card Brand Tokenization securely protects card information and promotes a possible higher conversion rate.

This feature is available for both credit and debit.

Lifecycle: Once a card token is generated, it will go through different stages throughout its existence; The Issuer can also request the replacement of the registration data of the physical card linked to that token, for example, for reasons of expiry, fraud, damage to the plastic, etc.

The Card Brand Token intermediated by Rede is currently only available for the Visa Card Brand.

Target Audience

This functionality is intended for e-commerce establishments.

Benefits

- Tokenization guarantees the protection of data in transit, as it consists of replacing sensitive card data with random numbers (cryptograms), called Card Brand Token.
- In the event of a data leak, your customers' cards remain safe, with no risk of fraudulent transactions.
- Greater card data security and increased reliability in transactions with tokens;
- It also guarantees compliance with regulatory standards laid down by law, such as the LGPD and PCI DSS (Data Security Standard).

First steps

The process of requesting card tokenization of Rede is done in two steps. First, the user sends the data to the card brand and receives a response with a unique identifier for that request. Then (in an asynchronous process) the card brand will send the token information, which will be updated in the registry, allowing its query by the cardholder.

Sending the card tokenization request to the card brand:

POST: /token-service/v1/tokenization

The e.Redre APIs use Basic Auth, which is an industry standard for the authorization and authentication of users and applications. This protocol is based on simplifying customer development for authorization flows for web, desktop apps, mobile phones, etc.

After contracting the Tokenization service on the Logged Portal “userede.com.br”, activation of the product on the Brand may take up to 3 days.

Rede customers must contract the service via the Logged Portal: Main menu "to sell"/ E-commerce option/ Rede Card Brand Tokenization/Contract.

Endpoints

Endpoints are the URLs that will be used to call a particular service. They may vary depending on the HTTP environment and method.

The composition is carried out as follows:

- Base URL
- API Version
- Service

Environment	URL
Sandbox	https://rl7-sandbox-api.useredecloud.com.br/v1/tokenization
Production	https://api.userede.com.br/redelabs/token-service/v1/tokenization

Registration of the URL to receive the Token Events known as the Lifecycle

After contracting the Card Brand Tokenization service and meeting the SLA deadline of up to 02 hours, as soon as the e.Redde Customer re-enters their logged-in area of the "userede.com.br" portal, the endpoint registration option will be available.

In order to complete this step in the configuration and receive all the notifications related to the Token Lifecycle, it is necessary to register the endpoint.

The Rede customer must contract the service via the logged Portal: Main menu → to sell → E-commerce option/ Rede Card Brand Tokenization/ Contract

Endpoint Registration for Authenticated and Unauthenticated URLs

The REDE must be informed by the Establishment whether the URL that is registered has authentication or not.

- URL Without Authentication:

Only register the valid and secure URL, so the fields: authorization/type: Bearer or Basic and authorization token are no longer required;

- URL With Authentication:

Once the authorization object has been passed, sending the fields: authorization/type Bearer or Basic and authorization token becomes mandatory;

Point of attention:

In the case of the Authenticated URL:

Select only one type of authentication: authorization/type (e.g. Bearer or Basic) and enter the Authorization token field to finish the process.

Tokenization Flow and Token Cryptogram

The process of requesting card tokenization of Rede is done in two steps. First, the Establishment sends the data that is sent to the card brand and receives a response with a unique identifier for that request, and then (in an asynchronous process) the card brand sends the token information that will be updated in the register, allowing its query by the cardholder.

Requisition parameters:

Name	Size	Type	Mandatory	Description
email	Up to 200	Alphanumeric	Yes	E-mail address of the cardholder, customer or commercial establishment
cardNumber	Up to 19	Alphanumeric	Yes	Token informed by the Card Brand. Identify tokenized transactions
expirationMonth	2	Alphanumeric	Yes	Card expiration month (between "01" and "12")
expirationYear	4	Alphanumeric	Yes	Card expiration year
cardholderName	Up to 200	Alphanumeric	No	Cardholder's name printed on the card.
securityCode	Up to 4	Alphanumeric	No	Card security code located on the back of the card
storageCard	Up to 2	Numeric	Yes	Indicates operations that may or may not be using COF (Card on File): 0 Transaction with credential not stored. 2 - Transaction with credential already stored.

Response parameters:

Name	Size	Type	Description
returnCode	Up to 3	Alphanumeric	Request return code
returnMessage	Up to 256	Alphanumeric	Request return message

tokenizationId	36	Alphanumeric	Unique identifier of the card tokenization request by REDE
----------------	----	--------------	--

Token query

The query of the tokenization request can be done right after obtaining the tokenizationId from Rede, it will always be done when the Establishment receives an event through the webhook, where it will be possible to see the status of the request and other information listed below.

Request to query the tokenization request data:

GET: /token-service/v1/tokenization/{tokenizationId}

Requisition parameters:

Name	Size	Type	Mandatory	Description
tokenizationId	36	Alphanumeric	Yes	Unique identifier of the card tokenization request by REDE

Response parameters:

Name	Size	Type	Description
returnCode	Up to 3	Alphanumeric	Request return code
returnMessage	Up to 256	Alphanumeric	Request return message
affiliation	Up to 9	Numeric	Establishment's affiliation number (PV)
tokenizationId	36	Alphanumeric	Unique identifier of the card tokenization request by REDE
tokenizationStatus	-	Alphanumeric	Status of the tokenization request: <ul style="list-style-type: none"> • Pending • Active

			<ul style="list-style-type: none"> • Inactive • Suspended • Failed
brand/name	-	Alphanumeric	Card brand name of the card's BIN sent in the tokenization request
brand/message	Up to 256	Alphanumeric	Return message from the card brand in the event of a failed token request for a specific card. The tokenizationStatus in this scenario will be Failed and the token information will not be available
lastModifiedDate	-	Datetime	Date of the last update of the record in the format YYYY-MM-DDThh:mm:ssTZD
last4	4	Alphanumeric	4 last digits of the card
token/code	Up to 16	Numeric	Number of the decrypted card token, generated by the card brand
token/expirationDate	7	Alphanumeric	Expiration date (in MM/YYYY format) of the token generated by the card brand for the card sent

Cryptogram

The crypto has a lifecycle and it is per transaction. Once used, it is not stored.

Every transaction needs a new cryptogram, so a request will be made whenever the customer confirms a tokenized transaction, which must be generated and sent with each new transaction, along with the token data

The provisioning of the token and the payment flow are different.

For a deleted or suspended token, if you request it, the card brand will display an error. A cryptogram cannot be requested for tokens with these statuses.

The cryptogram is for the exclusive use of that transaction and cannot be used in others, and it is not stored.

Note: the Visa card brand allows up to 6000 cryptograms to be requested for the same token within a 90-day window.

Cryptogram query

The cryptogram query is made by sending the tokenizationId returned in the tokenization request. The cryptogram data is only returned after confirming the creation of the card brand token, so at some point in the request lifecycle, an error may be displayed about the unavailability of the token's cryptogram (listed in the error section).

Request to query the cryptogram data of the token generated in the tokenization request:

POST: /token-service/v1/cryptogram/{tokenizationId}

Requisition parameters:

Name	Size	Type	Mandatory	Description
tokenizationId	36	Alphanumeric	Yes	Unique identifier of the card tokenization request by REDE
subscription	-	Boolean	No	<p>Informs the issuer whether a transaction is the result of a recurrence.</p> <p>If the transaction is a recurrence, send true. Otherwise, send false.</p> <p>Failure to submit this field will be considered the value false.</p>

Response parameters:

Name	Size	Type	Description
returnCode	Up to 3	Alphanumeric	Request return code
returnMessage	Up to 256	Alphanumeric	Request return message
tokenizationId	36	Alphanumeric	Unique identifier of the card tokenization request by REDE

cryptogramInfo/tokenCryptogram	28	Alphanumeric	Cryptogram of the token generated by the card brand in the process of requesting tokenization of the card. Value in Base64 format, with a maximum of 28 characters
cryptogramInfo/eci	2	Alphanumeric	Code returned by the Card Brands indicating the result of the cardholder's authentication with the Issuer.
cryptogramInfo/expirationDate**	24	Datetime	Expiration date (in AAAA-MM-DDThh:mm:ss.sssZ format) the cryptogram of the token generated by the card brand for the card sent

cryptogramInfo/expirationDate:** The token's cryptogram expiration date may not be returned by some card brands.

Callback from the webhook

After the tokenization request, each time the card brand is updated regarding the status of that request, it will be returned to the url provided by the establishment, sending the following information.

The Establishment must provide a valid and secure url, which must be called by the Tokenization process to receive notification of events (via a POST method), known as the Token Lifecycle."

This URL will be registered by CNPJ, regardless of how many or which PVs have been enabled for that establishment.

The Customer can only associate one URL for each CNPJ, and it is possible to delete or change it.

IMPORTANT: If the notification endpoint/ url is not entered, no event will be notified during the tokenization process.

Note: In the sandbox environment, this callback will be performed 2 minutes after the tokenization request is sent.

Requisition parameters:

Name	Shipping point	Size	Type	Description
Authorization	header	Up to 3	Alphanumeric	Header for authorizing the request at the url provided by the establishment via the logged portal.
Request-ID	header	Up to 36	Alphanumeric	Unique identifier of the request
Content-Type	header	-	Alphanumeric	Fixed value set to 'application/json'
id	body	6	Alphanumeric	Unique callback identifier
merchantId	body	9	Alphanumeric	Establishment's affiliation number (PV)
events	body	-	Alphanumeric list	Name of the events that will be reported to the customer. Example: ["PV.TOKE NIZACAO-BAND EIRA"]
data/tokenizationId	body	Up to 36	Alphanumeric	Unique identifier of the card tokenization request by REDE

Returns

Successful returns

By default, whenever a request is successfully made to Rede, the code and message described in the table below will be displayed:

Error code	Description
00	Success

Integration returns - Token Requestor

Integration returns are displayed whenever there is something wrong with your request, thus allowing immediate correction.

Error code	Description	HTTP Status Code
01	TokenizationId: Required parameter missing	400
02	TokenizationId: Invalid guid value	400
03	There is no data with the given guid	404
04	Email: Required parameter missing	400
05	Email: Invalid parameter size	400
06	Email: Invalid parameter format	400
07	CardNumber: Required parameter missing	400
08	CardNumber: Invalid parameter size	400
09	CardNumber: Invalid parameter format	400
10	ExpirationMonth: Required parameter missing	400
11	ExpirationMonth: Invalid parameter value	400
12	ExpirationMonth: Invalid parameter format	400
13	ExpirationYear: Required parameter missing	400
14	ExpirationYear: Invalid parameter size	400
15	ExpirationYear: Invalid parameter format	400
16	CardholderName: Required parameter missing	400
17	CardholderName: Invalid parameter size	400
18	CardholderName: Invalid parameter format	400
19	SecurityCode: Required parameter missing	400
20	SecurityCode: Invalid parameter size	400
21	SecurityCode: Invalid parameter format	400
22	Expired card	400

23	Service not enabled for this establishment	403
24	Affiliation: Invalid parameter size	401
25	Affiliation: Invalid parameter format	401
26	Affiliation: Required parameter missing	401
27	Card is from a brand not enabled for tokenization	403
28	Url: Required parameter missing	400
29	Service temporarily unavailable	503
30	Declined: This card is considered ineligible for tokenization at this moment	403
31	NotAllowed: This card is considered ineligible for tokenization at this moment	409
32	TokenCryptogram unavailable. Check the token status	403
33	Failed	-
34	Authorization Token: Required parameter missing	400
35	Authorization Token: Invalid parameter format	400
36	Authorization Type: Required parameter missing	400
37	Authorization Type: Invalid parameter format	400
38	Cryptogram token unavailable. Check the token status	400
39	"Service temporarily unavailable"	400
40	Authorization Type: Invalid parameter format	400
41	Declined: This card is considered ineligible for tokenization at this moment	400

42	NotAllowed: This card is considered ineligible for tokenization at this moment	400
43	NotAllowed: This affiliation is not valid (inactive/not found)	400
44	Internal error occurred. Please contact Rede	400
45	storageCard: Required parameter missing	400
46	storageCard: Invalid parameter format	400
47	subscription: Invalid parameter format	400
48	TokenizationStatus: Required parameter missing	400
49	tokenizationStatus : Invalid value	400
50	tokenizationStatus : Invalid value	400
51	Reason:Invalid value	400
65	Token: Required parameter missing	401
89	Token: Invalid token	401

Token Management

After creating the token, it is possible to manage its status. That is, the establishment has the autonomy to delete, suspend and reactivate the tokens under its responsibility.

PUT: token-service/v1/tokenization/{tokenizationId}

Requisition parameters:

Name	Size	Type	Mandatory	Description
tokenizationStatus	100	Alphanumeric	Yes	If the updated status of the token is to delete, it will be the deleted If the updated status of the token is to suspend, it

				will be the suspend If the updated status of the token is to reactivate, it will be the resume
Reason	2	Numeric	Yes	Reason for update: 1-Customer Request 2- suspected fraud

Parameter response

Name	Size	Type	Description
returnCode	Until 3	Alphanumeric	Request return code
returnMessage	Until 256	Alphanumeric	Request return message
tokenizationId	Until 36	Alphanumeric	Unique identifier of the card tokenization request by the Network
Brand/name*	-	Alphanumeric	Brand name. Ex: Visa
Brand/message*	-	Alphanumeric	Brand error message. Ex: Card not allowed

* These fields will only be filled in if an error is returned from the brand

Sandbox Tutorial

Tokenization of Rede Card Brand

Simulate request status

To simulate tokenization requests in different lifecycle statuses, you can send the predefined values in the table below in the cardNumber field at the time of the request, and then perform

the query from the generated tokenId.

cardNumber	tokenizationStatus
000000000000000008	Inactive
000000000000000009	Suspended
000000000000000010	Deleted

Simulate errors of business rule

For similar errors related to the API's business rules, it is necessary to send a tokenization request with a cardNumber value from the list below:

cardNumber	Error code	Error message
000000000000000001	23	Service not enabled for this establishment

Simulate cryptogram query errors

To simulate the possible errors presented by the card brand when requesting the cryptogram, it is necessary to perform the queries by entering one of the tokenId available in the table below:



tokenId	Error code	Error message
01	29	Service temporarily unavailable
02	30	Declined: This card is considered ineligible for tokenization at this moment
03	31	Not allowed: This card is considered ineligible for tokenization at this moment

04	32	TokenCryptogram unavailable. Check the token status
05**		-

***For the latter case, an error code/message will not be displayed, but the information in the cryptogramInfo/expirationDate field will be omitted.

Simulate errors of card brand

To simulate error messages of card brand, simply send tokenization requests a cardNumber value from the list below and perform the query with the tokenizationId that is generated:

cardNumber	Card Brand	returnCode	returnMessage	brand/message	HTTP Code	Status
000000000000000000000002	Visa	33	Failed	provisionDataExpired: The PAN information provided is considered stale	400	
000000000000000000000003	Visa	33	Failed	cardVerificationFailed: invalidfield	403	
000000000000000000000004	Visa	33	Failed	cardNotEligible: This card cannot be used for tokenization at this moment	403	
000000000000000000000005	Visa	33	Failed	cardNotAllowed: The requested action is not allowed for a given PAN	403	
000000000000000000000006	Visa	33	Failed	dedined: This card is considered not eligible for tokenization at this time	403	
000000000000000000000007	Visa	33	Failed	not Allowed: Further operations for this card are no longer allowed, Contact your bank to resolve this issue	409	

Simulate webhook configuration

Life Cycle

The cards will have "one-to-many" relationships with tokens, which means that a single card will be associated with multiple tokens.

The tokens are associated with a credit card number. A single card can have different tokens associated with it, however, each token is unique and specific to a particular Establishment.

When a card lifecycle event occurs, they are updated with the new card's information. This brings fluidity to the Customer journey, making lifecycle management one of the main benefits of tokenization.

The status of each token for a given card is also independent.

One of the token's main functions is to control its lifecycle, which has 4 different statuses at the moment: Active, Inactive, Suspended and Deleted

- Requested: this is the initial state of a token that has already been created, but not yet made available and not functional.
- Active: when the token is available and can already be used
- Suspended: the token is temporarily suspended by the issuer, consumer or cardholder.
- Deleted: the token is permanently disabled and can no longer be used.

The registration of the url for receiving notifications follows the definitions shown in the tables below.

Note: In a production environment, this request will be made through the logged portal. Therefore, this example is to **guide the use/simulation of the resource in the Sandbox environment.**

Registration of the callback url for the notification webhook:

GET: /token-service/v1/tokenization/seturl

Requisition parameters:

Name	Size	Type	Mandatory	Description
url	Up to	Alphanumeric	Yes	Callback url for sending information to the tokenization

Name	Size	Type	Description
returnCode	Up to 3	Alphanumeric	Request return code
returnMessage	Up to 256	Alphanumeric	Request return message
	256		requester via Webhook
authorization/type	Alphanumeric	No**	Type of authorization to be performed on the callback url provided by the establishment. The possible values are: <ul style="list-style-type: none"> • Bearer • Basic
authorization/token	Alphanumeric	No**	Token to be used in the authorization process at the callback url provided by the establishment. It must be sent in the format 'Bearer XXX' or 'Basic XXX' according to the type provided in the previous field, where XXX will be the token itself.

**It should be noted that once the authorization object has been passed, sending the type and token fields becomes mandatory

Response parameters:

Token management

PUT: (token-service/v1/tokenization/{tokenizationId})

Simulate request status

To simulate tokenization requests in different lifecycle statuses, you can send the predefined values in the table below in the cardNumber field at the time of the request, and then perform the query from the generated tokenizationId

Request parameters

Name	Size	Type	Mandator y	Description
tokeniza tionStat us	100	Alphanu meric	Yes	If the updated status of the token is to delete, it will be the deleted If the updated status of the token is to suspend, it will be the suspend If the updated status of the token is to reactivate, it will be the resume
Reason	2	Numeric	Yes	Reason for update: 1-Customer Request 2- suspected fraud

Response Parameters

Name	Size	Type	Description
returnCode	Until 3	Alphanumeric	Request return code
returnMessage	Until 256	Alphanumeric	Request return message
tokenizationId	Until 36	Alphanumeric	Unique identifier of the card tokenization request by the Network
Brand/name*	-	Alphanumeric	Brand name. Ex: Visa
Brand/message*	-	Alphanumeric	Brand error message. Ex: Card not allowed

Simulate token management query errors

To simulate error presented by card brand upon doing token management requests it is necessary to send the requests containing one of the tokenizationId available below:

PUT: (token-service/v1/tokenization/{tokenizationId})

Nome	Tamanho	Tipo	Obrigatório	Descrição
tokenizationStatus	100	Alphanumeric	Yes	<p>If the updated status of the token is to delete, it will be the deleted.</p> <p>If the updated status of the token is to suspend, it will be the suspend.</p> <p>If the updated status of the token is to reactivate, it will be the resume.</p>
reason	2	Numérico	Sim	<p>Reason for update:</p> <p>1-Customer Request</p> <p>2- suspected fraud</p>

Parâmetros de Resposta

Nome	Tamanho	Tipo	Descrição
returnCode	Up to 3	Alphanumeric	Request return code
returnMessage	Up to 256	Alphanumeric	Request return message
tokenizationId	36	Alphanumeric	Unique identifier for the card tokenization request through Rede
Brand/name*	-	Alphanumeric	Brand name. Example: Visa
Brand/message*	-	Alphanumeric	Brand error message. Example: Card not allowed

Recurrence and Card-on-File

What is Recurrence?

Recurrence is a payment option that works like a periodical charge, payment is made for a period and frequency predetermined by the merchant and agreed with the cardholder. The main advantage is not compromising the customer's card limit, making charges automatically. It is an option that helps the retailer not to worry about charging customers frequently, since this adjustment has already been made at the time of purchase.

We remind you that e.Redes does not have a recurrence engine or manages recurring appointments, if your e-commerce has this engine, you must send recurring transactions to Rede using the fields correctly

What is card-on-file?

Also known as credential-on-file or stored credential. Stored card transactions are those in which the cardholder has authorized the storage of their card data to initiate future purchases or that can be used by the merchant for recurrences or future charges.

When we have a stored card, it is not necessary to send the securityCode in the transactions.

Check below which fields to use in each operation:

Recurring transactions	Card-on-file
<ul style="list-style-type: none"> • storagecard • credentialId (para Mastercard) • subscription • brandTid 	<ul style="list-style-type: none"> • storagecard • credentialId (para Mastercard)

What do each of them mean?

- **storagecard:** Indicates operations that may or may not be using COF (Card On File – stored credential)
 - 0- Used for non-stored card (must be accompanied by securityCode)
 - 1- Used for a card that is being stored for the first time must be accompanied by the securityCode, if this is the first transaction in which the card is to be stored. This step can be replaced by a Zero Dollar transaction, which must also be accompanied by the security code.)
 - 2- Used to indicate card already stored (securityCode must not be sent)

For the Elo brand it is mandatory that before indicating stored card (storageCard=2) a Zero Dollar or transaction with storagecard = 1 has been made

Important: Rede will only signal a transaction with a stored credential when the storageCard is equal to “2”, transactions signaled as 1 indicate that the merchant is executing a transaction that will initiate the storage of the credential, thus requiring validations of common transactions. EX: use of card security code (securityCode).

- **credentialId:** Field currently used only by the Mastercard brand, to indicate the reason for storing the card and facilitate analysis for approval. For more details see the following section "[Categorizing card-on-file transactions](#)"
- **subscription:** Parameter used to signal recurring transactions. It must be sent as true if the transaction is recurrent, false for common transactions;
- **brandTid:** This is a unique and dynamic field, received with each transaction response. It is used to correlate recurrence plans (in the case of the Visa and Mastercard brands) and to correlate transactions initiated by the merchant, such as incremental, recurring, and stored card transactions (in the case of the Elo brand).

It must be sent to the brand from the second transaction, and it correlates subsequent transactions with the first.

The merchant is responsible for storing the returned brandTid and sending it in all subsequent transactions.

For the Visa brand: When the transaction is identified as recurring (subscription=true), each subsequent transaction that the merchant sends to that recurrence plan must also send the brandTid provided by the brand of the original transaction.

For example: In a monthly recurring plan, in the third transaction the merchant must forward the brandTid received in the transaction that started the recurring plan.

In the case of Visa tokenized transactions, sending this parameter is mandatory. In other types of transactions, if done, it must be ensured that the correct amount is sent to avoid denials by the brand.

For the Mastercard brand: When the transaction is identified as recurring (subscription=true), each subsequent transaction that the merchant sends to that recurrence plan must also send the brandTid provided by the brand of the original or previous transaction.

For the Elo brand: Whenever the establishment initiates a transaction, whether by recurrence, stored card, or any other types of incremental transactions, it must send the brandTid provided by the brand in the original transaction.

For Elo, if a Zero Dollar validation was carried out before using the card, the brandTid received at Zero Dollar must be sent in subsequent financial transactions.

Remember: The establishment will continue to receive a different brandTid from the brand in each new transaction, but when requesting authorization for a recurring transaction or one initiated by the establishment, it must follow the rules described above. Pay attention to the formats and always ensure that the parameter is sent correctly to avoid denials by the issuer/brand.

If you do not have the corresponding value, we recommend starting a new card storage process (card-on-file) with the cardholder to obtain the parameter to be sent in subsequent transactions.

Pay attention to the points below:

- Failure to send the storagecard field will be considered 0 (credential not stored).
- If the customer wants to change the card, it will be necessary to restart the process from sending the first transaction, that is, storageCard=1 for the new card, then in the others, brandTid and storageCard=2 can be sent, in addition to the indication of recurrence (subscription=true).
- We emphasize that recurring transactions **cannot be processed as pre-authorization**, for this the **capture** field must be equal to “true”, indicating an automatic capture. **If they are sent as capture “false”, the recurrence marking will not be considered.**
- In addition, any change in recurrence (eg change in the amount billed monthly) will be considered a new transaction and the old one will be disregarded.
- When carrying out transactions using a stored credential (card-on-file), if the merchant already wants to start the charges or just save the card for future charges, the brands **demand** that a Zero Dollar validation is carried out beforehand.
- The Zero Dollar validation can be used by merchants to verify card data to check if the credential is valid and can be stored, in addition to allowing future charges without sending the securityCode in recurring charges with stored credential or only stored credential, in addition to increasing the possibility of conversion.
- **Use of “sai” in card-on-file transactions (stored credential):** The parameter must be used whenever the transaction has a specific ECI, which is not linked to 3DS authentication (ex: Wallets and Cloud Token Visa). When authenticated as 3DS, it is necessary that the “eci” is informed within the 3D Secure group, not being necessary to use the “sai” in this case.
- When sending the threeDSecure group in any request, the “sai” field will be ignored and the “eci” from the threeDSecure group will be prioritized.
- When using recurrence with Elo, the card brand demands that before the first transaction, a Zero Dollar authentication should be done.

Categorizing of card-on-file transactions

Since October 2022, due to brand regulatory changes, Mastercard card-on-file transactions will be categorized into 12 types of categories CIT (Cardholder Initiated - Card Holder) and MIT (Initiated by Merchant – Merchant). Since June 1st, 2023, Mastercard started monitoring the field usage, be warned as this can result on compliance actions.

The continued growth of e-commerce, along with the increase in transaction types, necessitates the need to understand consumer intent. The introduction of the CIT or MIT indicator provides transparency allowing use for:

- Issuer authorization
- Logic fraud detection
- Dispute management

Therefore, it will be necessary to adjust your integration with e.Redes to send a new field called "credentialId", which will be part of the "transactionCredentials" group. Thus, when storageCard is equal to 1 or 2, indicating that the card is being or has already been stored, it will be **mandatory** to indicate in which category the card-on-file transaction (stored credential) falls.

This field must be sent in Zero Dollar requests that intend to store the card information.

Sending this field has been mandatory for Mastercard operations since June 1, 2023, and as of June 1st, 2024, Mastercard may apply penalties in case of non-conformity from merchants, relative to the period out of the norm. Among the benefits of sending this field is the ability to support the brand and the issuer in the analysis of their transactions, which can help in the conversion. The other fields currently used for similar purposes, such as storagecard, subscription and installments, need to continue to be populated.

The following table lists the categories to be considered:

Main categories	Indicator to be sent	Corresponding Sub-Category	Definition	Example
IND. C1 (Cat) Cardholder-Initiated Any transaction where the cardholder is actively participating in the transaction. Transactions can be performed based on credentials provided by the cardholder at the	01	Credential on File	The consumer agrees that their card will be stored with the merchant for future transactions that may occur from time to time.	Car app transactions.
	02	Standing Order	The consumer agrees that their card is stored with the	Monthly payment for services.

<p>time of the transaction or a credential stored on file from a previous interaction.</p> <p>Transactions can take place as an in-store POS transaction, an e-commerce transaction, a mail order/phone order transaction, or at an ATM</p>			<p>merchant and initiates the first transaction in a series intended for a variable amount and a fixed frequency.</p>	
	03	Subscription	<p>The consumer agrees that their card is stored and initiates the first transaction in a series destined for a fixed value and a fixed frequency</p>	<p>Monthly newspaper subscription</p>
	04	Installments	<p>The consumer agrees that their card is stored to establish a plan of installment payment and initiates the first transaction in a series.</p>	<p>Installments Transactions</p>
<p>IND. M1 (Cat) Merchant-Initiated Recurring Payment or Installment</p> <p>An operation arising from an agreement between the cardholder and the merchant in which the cardholder agrees for the merchant to store the credential cardholder's data and to use that stored credential on file for a later purchase of goods or services.</p>	05	Unscheduled Credential on File	<p>Transactions carried out by an agreement between a cardholder and a merchant, whereby the consumer authorizes the merchant to store and use the cardholder's account data to initiate one or more future transactions.</p>	<p>Toll Auto recharge</p>
	06	Standing Order	<p>Use cardholder account details for a transaction that must take place at regular intervals for a variable amount</p>	<p>Monthly service payments</p>
	07	Subscription	<p>Use cardholder account details for a transaction that must take place at regular intervals for a fixed amount.</p>	<p>Monthly subscription or fixed monthly service payment.</p>
	08	Installments	<p>Store cardholder account data for merchant use to initiate one or more future transactions for a known amount</p>	<p>Buy a TV for BRL 1,000, paying in four equal installments of BRL 250 (the first transaction is CIT, the remaining three transactions are MIT).</p>

			with a specified duration based on a single purchase.	
IND. M2 (Cat) M2 Merchant-Initiated Industry Practice A transaction initiated by the merchant to fulfill a business practice that occurs most frequently after an initial interaction with the cardholder. Industry practice transactions may be carried out with credentials stored on file or credentials that are not stored on file but are temporarily held by the merchant as agreed by the consumer.	09	Partial Shipment	Occurs when an agreed quantity of goods ordered by e-commerce is not available for shipment at the time of purchase. Each shipment is a separate transaction	The consumer has ordered goods that are shipped at different times.
	10	Related/Delayed Charge	An additional charge to the account after initial services have been provided and the Payment has been processed.	The consumption charge after the cardholder does the check-out at an hotel.
	11	No show	A penalty charged in accordance with the merchant's cancellation policy.	Cancellation of a reservation by the cardholder without adequate notice to the merchant.
	12	Reenvio	Attempting to obtain authorization for a transaction that was declined, but the issuer's response does not prohibit the merchant from trying later.	Insufficient funds/response over credit limit/transit retry

Returns

In order to improve the declined transaction response visualization, Rede has fields which display the brand's complete deny reason:

- *Brand* group (ABECS standard);

Important: the use of the *brand* group is mandatory, as it provides the detailed cause for transaction denial.

Another possible case is by use of the old Rede encapsulated codes, see [Issuer Returns](#):

- Return code – outside Brand group (Rede encapsulated code);
- Return message – outside Brand group (Rede encapsulated code);

Important: The use of the old Rede encapsulated code is not recommended as it doesn't provide enough information on declined transaction reason, and will soon be discontinued.

Brand Returns

As of July 15, 2020 we started to offer our customers the option of receiving open brand codes, sent by banks, depending on the reason for denying the transaction.

Along with this option, we started to comply with Abecs regulation 21, which standardizes messages in relation to these transactions denied in the authorization process.

The goal is to provide greater transparency and standardization, seeking to increase the approval rate.

Besides the standardized return for the main brands (ELO, Visa, Master/Hiper and Amex), it will be possible, through the table below, to verify whether that refusal is reversible or irreversible. Very important to the retry process, so pay attention to the codes returned.

For the other card brands, the transaction denied messages remain the same, however, with open source.

It is worth adding that in order to gain access to open return codes with the standardization of the message, it is necessary to make a small adjustment in your API, follow the activation procedure below.

If you do not make this adjustment, current returns remain at the Rede standard, without any change or impact.

Activation:

To further enable this functionality and start receiving messages standardized by the ABECs regulations and other returns from the brand with open codes, simply adjust the custom header field, "Transaction-Response", with the value "brand-return-opened " filled in, this is true for both the transaction and the query.

Header	Value
Transaction-Response	brand-return-opened

In this way, the response in case of an approved transaction starts to return the "Brand" object with the fields of the brand, "authorizationCode" and "brandTid", and additionally "Name", "returnCode" and "returnMessage". In case of a denied transaction, the "Brand" object will only have the "Name", "returnCode" and "returnMessage" fields.

If the header "Transaction-Response" with the value "brand-return-opened" is not sent in the transaction, it can be sent normally in the query and the information about the brand (Brand) will be returned.

The returnCode and returnMessage fields from outside the "Brand", which are the ones we know today, now have only the return codes of transactions denied on the Rede, again.

Brand return code table and message standardization:

Reason	ELO	Visa	Master/Hiper	Amex	Message	E-commerce Message
Generic	5 - reversible	5 - reversible	5 - reversible	100 - reversible	Contact your card center	Please contact issuer
Balance/insufficient limit	51 - reversible	51 - reversible	51 - reversible	116 - reversible	Unauthorized	Refused
Invalid password	55 - reversible	55 - reversible 86 - reversible	55 - reversible 86 - reversible	117 - reversible	Invalid password	Invalid pin
Transaction not allowed for the card	57 - irreversible	57 - irreversible	57 - reversible	200 - irreversible	<p>Visa and Elo: Transaction not permitted to cardholder. Do not retry</p> <p>Amex: unauthorized transaction. Do not try again</p> <p>Amex: unauthorized transaction. Do not try again</p> <p>Mastercard: Transaction not permitted to cardholder.</p>	<p>Visa and Elo: Transaction not permitted to cardholder. Do not retry</p> <p>Amex: unauthorized transaction. Do not try again</p> <p>Mastercard: Transaction not permitted to cardholder.</p>
Card number does not belong to the	14 - irreversible 56 - irreversible	14 - irreversible	14 - irreversible 1 - irreversible	122 - irreversible	Check your card data	<p>Format error. Verify card data.</p> <p>Visa: invalid card. Do not retry</p>

Reason	ELO	Visa	Master/Hiper	Amex	Message	E-commerce Message
issuer invalid card number.						
Security breach	63 - irreversible	N7 - irreversible	63 - reversible	122 - irreversible	Check your card data	Format error. Verify card data.
Invalid or not present						Visa: invalid card. Do not retry
Suspected Fraud / Travel notice	59 - reversible	59 - reversible	63 - reversible	100 - reversible	Contact your card center	Please contact issuer
Invalid merchant	58 - irreversible	3 - irreversible	3 - irreversible	109 - irreversible	Transaction not allowed - do not try again	Unauthorized transaction. Do not try again
Redo the transaction (issuer requests retry)	4 - reversible	No corresponding code	No corresponding code	No corresponding code	Redo the transaction	Please, retry this transaction.
Consult accreditor	6 - reversible	No corresponding code	No corresponding code	No corresponding code	Merchant, contact the acquirer	Contact card issuer
Problem at the acquirer	19 - irreversible	19 - irreversible	30 - irreversible	No corresponding code	Card error – do not try again	Invalid card. Do not retry
Card error	12 - irreversible	6 - irreversible	No corresponding code	115 - irreversible	Check your card data	Format error. Verify card data Visa: invalid card. Do not retry Amex: function not supported. Do not retry

Reason	ELO	Visa	Master/Hiper	Amex	Message	E-commerce Message
Format error (messaging)	30 - irreversible	12 - irreversible	30 - irreversible	181 - irreversible	Card error – do not try again	Invalid card. Do not retry
Invalid transaction amount	13 - irreversible	13 - irreversible	13 - irreversible	110 - irreversible	Transaction amount not allowed - do not try again	Transaction amount not permitted. Do not retry
Invalid installment value	23 - irreversible	No corresponding code	12 - irreversible	115 - irreversible	Invalid installment - do not try again	Function not supported. Do not retry
Exceeded password attempts purchases	38 - reversible	75 - reversible	75 - reversible	106 - reversible	Exceeded password attempts. Contact your card center	Invalid pin. Contact card issuer
Lost Card	41 - irreversible	41 - irreversible	41 - irreversible	200 - irreversible	Transaction not allowed - do not try again	Unauthorized transaction. Do not try again
Stolen Card	43 - irreversible	43 - irreversible	43 - irreversible	200 - irreversible	Transaction not allowed - do not try again	Unauthorized transaction. Do not try again
Card expired / invalid expiration date	54 - irreversible	54 - irreversible	54 - irreversible	101 - irreversible	Check your card data	Format error. Verify card data. Visa: invalid card. Do not retry

Reason	ELO	Visa	Master/Hiper	Amex	Message	E-commerce Message
Transaction not allowed terminal capacity	57 - irreversible	58 - irreversible	58 - irreversible	116 - irreversible	Transaction not allowed for card - do not try again	Transaction not permitted to cardholder. Do not retry Amex: refused
Excess amount withdrawal	61 - reversible	61 - reversible N4 - reversible	61 - reversible	No corresponding code	Value exceeded. Contact your card center	Transaction amount not permitted. Contact card issuer
Temporary block (ex: default)	62 - reversible	62 - reversible	57 - reversible	No corresponding code	Contact your card center	Contact card issuer
Invalid minimum transaction amount	64 - irreversible	No corresponding code	13 - irreversible	No corresponding code	Transaction amount not allowed - do not try again	Transaction amount not permitted. Do not retry
Exceeded withdrawal amount.	65 - reversible	65 - reversible	65 - reversible	No corresponding code	Exceeded withdrawal amount. Contact your card center	Exceeds withdrawal limit. Contact card issuer
Expired password / password encryption error	83 - irreversible	74 - irreversible 81 - irreversible	88 - irreversible	180 - irreversible	Invalid password - do not try again	Invalid pin. Contact card issuer
Exceeded password attempts/withdrawal	75 - reversible	75 - reversible	75 - reversible	106 - reversible	Exceeded password attempts.	Invalid pin. Contact card issuer

Reason	ELO	Visa	Master/Hiper	Amex	Message	E-commerce Message
					Contact your card center	
Invalid or non-existent target account	76 - irreversible	No corresponding code	No corresponding code	No corresponding code	Invalid target account - do not try again	Format error. Do not try again (invalid account)
Invalid or non-existent source account	77 - irreversible	No corresponding code	No corresponding code	No corresponding code	Invalid source account - do not try again	Format error. Do not try again (invalid account)
New card without unlocking Includes card blocked by the customer in the app (ecom, nfc)	78 - reversible	78 - reversible	57 - reversible	No corresponding code	Unlock the card	card not initialized
Invalid card (cryptogram)	82 - irreversible	82 - irreversible	88 - irreversible	180 - irreversible	Card error – do not try again	Invalid card. Do not retry Master/Hiper: invalid pin. Contact card issuer Amex: invalid pin. Contact card issuer
Issuer off the air	91 - reversible	91 - reversible	91 - reversible	912 - reversible	Communication failure - try again later	Error. Retry transaction
System failure	96 - reversible	96 - reversible	96 - reversible	911 - reversible	Communication failure - try again later	Error. Retry transaction

Reason	ELO	Visa	Master/Hiper	Amex	Message	E-commerce Message
Difference - pre authorization	No corresponding code	N8 - irreversible	No corresponding code	No corresponding code	Different pre-authorization value - do not try again	Format error. Do not retry (authorization amount differs)
Incorrect function (debit)	Ab - reversible	52 - reversible 53 - reversible	No corresponding code	No corresponding code	Use credit function	Not supported. Submit transaction as credit
Incorrect function (credit)	Ac – reversible	39 - reversible	No corresponding code	No corresponding code	Use debit function	Not supported. Submit transaction as debit
Incorrect function (voucher)	Av – reversible	No corresponding code	No corresponding code	No corresponding code	Use voucher function	Not supported. Submit transaction as voucher.
Password change / unlock	P5 - irreversible	No corresponding code	No corresponding code	No corresponding code	Invalid password - do not try again	Invalid pin. Contact card issuer
New password not accepted	P6 - reversible	No corresponding code	55 - reversible	No corresponding code	Invalid password use new password	Invalid pin. Contact card issuer
Collect card	No corresponding code	4 - irreversible	4 - irreversible	No corresponding code	Contact your card center - do not try again	Contact card issuer. Do not retry
Dynamic key change error	No corresponding code	N7 - irreversible	No corresponding code	No corresponding code	Card error – do not try again	Invalid card. Do not retry

Reason	ELO	Visa	Master/Hiper	Amex	Message	E-commerce Message
Confirmed Fraud	57 - irreversible	7 - irreversible	4 - irreversible	200 - irreversible	Transaction not allowed for card - do not try again	Transaction not permitted to cardholder. Do not retry Amex:unauthorized transaction. Do not try again
Issuer not found - incorrect bin (negative buyer)	No corresponding code	15 - irreversible	15 - irreversible	No corresponding code	Invalid card data - do not try again	Invalid card number. Do not retry
Failure to comply with anti-money laundering laws	No corresponding code	64 - irreversible	No corresponding code	No corresponding code	Contact your card center - do not try again	Contact card issuer. Do not retry
Invalid reversal	No corresponding code	76 - irreversible	No corresponding code	No corresponding code	Contact your card center - do not try again	Contact card issuer. Do not retry
Not found by router	No corresponding code	92 - irreversible	92 - irreversible	No corresponding code	Contact your card center - do not try again	Contact card issuer. Do not retry
Transaction denied for violation of law	57 - irreversible	93 - irreversible	62 - irreversible	No corresponding code	Transaction not allowed for card - do not try again	Transaction not permitted to cardholder. Do not retry

Reason	ELO	Visa	Master/Hiper	Amex	Message	E-commerce Message
Duplicate tracing data value	No corresponding code	94 - irreversible	94 - irreversible	No corresponding code	Contact your card center - do not try again	Contact card issuer. Do not retry
Surcharge not supported	No corresponding code	B1 - reversible	No corresponding code	No corresponding code	Contact your card center	Please contact issuer
Surcharge not supported by debit network	No corresponding code	B2 - reversible	No corresponding code	No corresponding code	Contact your card center	Please contact issuer
Force stip	No corresponding code	N0 - reversible	No corresponding code	No corresponding code	Contact your card center	Please contact issuer
Declined by CVV2 failure	No corresponding code	N7 – Reversible	No corresponding code	No corresponding code	Failure in CVV2 authentication	Declined by CVV2 failure
Withdrawal not available	No corresponding code	N3 - irreversible	No corresponding code	No corresponding code	Withdrawal not available - do not try again	Withdrawal not permitted. Do no retry
Recurring payment suspension for a service	No corresponding code	R0 - irreversible	No corresponding code	No corresponding code	Recurring payment suspension for service - do not retry	Recurring payment not permitted. Do not retry
Recurring payment suspension for all services	No corresponding code	R1 - irreversible	No corresponding code	No corresponding code	Recurring payment suspension for	Recurring payment not permitted. Do not retry

Reason	ELO	Visa	Master/Hiper	Amex	Message	E-commerce Message
					service - do not retry	
Transaction not eligible for visa pin	No corresponding code	R2 - irreversible	No corresponding code	No corresponding code	Transaction not allowed for card - do not try again	Transaction not permitted to cardholder. Do not retry
Suspension of all authorization orders	No corresponding code	R3 - irreversible	No corresponding code	No corresponding code	Recurring payment suspension for service - do not retry	Recurring payment not permitted. Do not retry
Account closed	46- Irreversible	46 – irreversible	62 – irreversible	No corresponding code	Transaction not allowed for card - do not try again	Transaction not allowed for card - do not try again
Id validation failure	No corresponding code	6P - irreversible	No corresponding code	No corresponding code	Id validation failure	Id validation failure
Use the chip	FM - irreversible	No corresponding code	No corresponding code	No corresponding code	Use the chip	Use the chip
Security Fraud	79 – irreversible (consult MAC)	Sem código correspondente	Sem código correspondente	Sem código correspondente	Unauthorized Transaction	Unauthorized Transaction

Other Returns

The returns below are for the exclusive use of the brands and are not within the 21 ABECS Regulation, but may occur in cases that the issuer/brand applies.

Brand	Code	Message	e-commerce Message
Mastercard	1	Please contact issuer	Please contact issuer
Mastercard	70	Please contact issuer	Please contact issuer
Mastercard	76	“To Account” specified Invalid/non-existent	Invalid account. Do not try again
Mastercard	77	“From Account” specified Invalid/non-existent	Invalid account. Do not try again
Mastercard	78	Invalid/non-existent account specified (general)	Invalid account. Do not try again
Mastercard	84	Invalid Authorization Lifecycle	Invalid Authorization Lifecycle
Mastercard	89	Password Unacceptable — Transaction Declined — Try Again	Invalid PIN. Try again
Visa	1	Please contact issuer	Please contact issuer
Visa	2	Please contact issuer	Please contact issuer
Visa	60	Verification failed [Cardholder ID does not match issuer records]"	Please contact issuer
Visa	70	PIN required	PIN data required
Visa	80	No financial impact	No financial impact
Visa	85	There is no reason to refuse a request for address verification, CVV2 verification, or proof of credit or merchandise return	Please contact issuer
Visa	1A	Additional client authentication required	Unauthorized transaction, try again
Visa	P5	PIN Unlock Denied - PIN change or unlock request denied by issuer	Invalid PIN. Do not retry
Visa	P6	PIN change denied - PIN requested insecure	Invalid PIN. Do not retry
Amex	107	Please contact issuer	Please contact issuer
Amex	111	Invalid Account	Invalid Account
Amex	121	Limit exceed	Limit exceed
Amex	122	Card security code printed with invalid key	Format error. Verify card data
Amex	130	Strong authentication required	Unauthorized transaction, try again
Amex	190	National Identification Incompatibility	Unauthorized transaction. Do not try again

Brand	Code	Message	e-commerce Message
Amex	191	Voice Referral	Please contact issuer
Amex	900	Advice accepted	Please contact issuer

In case of violation of the guidelines of the Brand retry program, the REDE will present an integration return code according to the classification of the retry. The return codes will be:

- 3025
- 3026
- 3027

See the table of [Integration Returns](#).

From July 7th 2023 onward, Rede will no longer use the Integration Return codes 3025, 3026, 3027 and instead will use the return codes below in two cases:

1. Return codes – inside *brand* group (ABECS)

Code	Message	Como atuar
N01	Declined by Rede: Issuer will never approve	Analyze negative return codes, stop excessive retries, whether from the cardholder or from automatic billing processes, and revisit your stored card base
N02	Declined by Rede: Excessive Reattempts	
N03	Declined by Rede: Attention – verify your Data	
N04	Declined by Rede: Subseller is not allowed to operate	Consult the call center Rede to assess the registration status of the sub-establishment
N99	Declined by Rede: Contact us	Contact the Rede as something in the operation needs to be reviewed and evaluate our Security Advice

2. Issuer return codes – outside *brand* group (non-ABECS)

Code	Message
124	Unauthorized. Contact Rede

Important: For perfect visualization of the decline reason codes from Rede's tool, the use of ABECS codes is necessary. For use instructions, see Brand Fees.

If the adjustment in the custom header is not done in order to enable the ABECS standard return codes, transactions will receive a response with a Issuer Return (124).

To know more about the Transaction reattempts program and its rules, see Brand Fees.

Card Center Returns

Issuer returns are displayed when a response is obtained from a credit or debit transaction request.

returnCode	returnMessage
00	Success
101	Unauthorized. Problems on the card, contact the issuer.
102	Unauthorized. Check the situation of the store with the issuer.
103	Unauthorized. Please try again.
104	Unauthorized. Please try again.
105	Unauthorized. Restricted card.
106	Error in issuer processing. Please try again.
107	Unauthorized. Please try again.
108	Unauthorized. Value not allowed for this type of card.
109	Unauthorized. Nonexistent card.
110	Unauthorized. Transaction type not allowed for this card.
111	Unauthorized. Insufficient funds.
112	Unauthorized. Expiry date expired.
113	Unauthorized. Identified moderate risk by the issuer.
114	Unauthorized. The card does not belong to the payment network.
115	Unauthorized. Exceeded the limit of transactions allowed in the period.

returnCode	returnMessage
116	Unauthorized. Please contact the Card Issuer.
117	Transaction not found.
118	Unauthorized. Card locked.
119	Unauthorized. Invalid security code
121	Error processing. Please try again.
122	Transaction previously sent
123	Unauthorized. Bearer requested the end of the recurrences in the issuer.
124	Unauthorized. Contact Rede
170	Zero dollar transaction not allowed for this card.
172	CVC2 required for Zero Dollar Transaction.
174	Zero dollar transaction success.
175	Zero dollar transaction denied.

Integration returns

Integration returns are displayed whenever there is something wrong with your request, thus allowing immediate correction.

returnCode	returnMessage
1	expirationYear: Invalid parameter size
2	expirationYear: Invalid parameter format
3	expirationYear: Required parameter missing
4	cavv: Invalid parameter size
5	cavv: Invalid parameter format
6	postalCode: Invalid parameter size

returnCode	returnMessage
7	postalCode: Invalid parameter format
8	postalCode: Required parameter missing
9	complement: Invalid parameter size
10	complement: Invalid parameter format
11	departureTax: Invalid parameter format
12	documentNumber: Invalid parameter size
13	documentNumber: Invalid parameter format
14	documentNumber: Required parameter missing
15	securityCode: Invalid parameter size
16	securityCode: Invalid parameter format
17	distributorAffiliation: Invalid parameter size
18	distributorAffiliation: Invalid parameter format
19	xid: Invalid parameter size
20	eci: Invalid parameter format
21	xid: Required parameter for Visa card is missing
22	street: Required parameter missing
23	street: Invalid parameter format
24	affiliation: Invalid parameter size
25	affiliation: Invalid parameter format
26	affiliation: Required parameter missing
27	Parameter cavv or eci missing
28	code: Invalid parameter size

returnCode	returnMessage
29	code: Invalid parameter format
30	code: Required parameter missing
31	softdescriptor: Invalid parameter size
32	softdescriptor: Invalid parameter format
33	expirationMonth: Invalid parameter format
34	code: Invalid parameter format
35	expirationMonth: Required parameter missing
36	cardNumber: Invalid parameter size
37	cardNumber: Invalid parameter format
38	cardNumber: Required parameter missing
39	reference: Invalid parameter size
40	reference: Invalid parameter format
41	reference: Required parameter missing
42	reference: Order number already exists
43	number: Invalid parameter size
44	number: Invalid parameter format
45	number: Required parameter missing
46	installments: Not correspond to authorization transaction
47	origin: Invalid parameter format
48	brandTid: Invalid parameter size
49	The value of the transaction exceeds the authorized
50	installments: Invalid parameter format

returnCode	returnMessage
51	Product or service disabled for this merchant. Contact Rede
53	Transaction not allowed for the issuer. Contact Rede.
54	installments: Parameter not allowed for this transaction
55	cardHolderName: Invalid parameter size
56	Error in reported data. Try again.
57	affiliation: Invalid Merchant
58	Unauthorized. Contact issuer.
59	cardHolderName: Invalid parameter format
60	street: Invalid parameter size
61	subscription: Invalid parameter format
63	softdescriptor: Not enabled for this merchant
64	Transaction not processed. Try again
65	token: Invalid token
66	departureTax: Invalid parameter size
67	departureTax: Invalid parameter format
68	departureTax: Required parameter missing
69	Transaction not allowed for this product or service.
70	amount: Invalid parameter size
71	amount: Invalid parameter format
72	Contact issuer.
73	amount: Required parameter missing
74	Communication failure. Try again

returnCode	returnMessage
75	departureTax: Parameter should not be sent for this type of transaction
76	kind: Invalid parameter format
78	Transaction does not exist
79	Expired card. Transaction cannot be resubmitted. Contact issuer.
80	Unauthorized. Contact issuer (Insufficient funds)
82	Unauthorized transaction for debit card.
83	Unauthorized. Contact issuer.
84	Unauthorized. Transaction cannot be resubmitted. Contact issuer.
85	complement: Invalid parameter size
86	Expired card
87	At least one of the following fields must be filled: tid or reference
88	Merchant not approved. Regulate your website and contact the Rede to return to transact.
89	token: Invalid token
97	tid: Invalid parameter size
98	tid: Invalid parameter format
99	BusinessApplicationIdentifier: Invalid parameter format.
100	WalletId: Invalid parameter format.
132	DirectoryServerTransactionId: Invalid parameter size.
133	ThreedIndicator: Invalid parameter value.
150	Timeout. Try again
151	installments: Greater than allowed
153	documentNumber: Invalid number

returnCode	returnMessage
154	embedded: Invalid parameter format
155	eci: Required parameter missing
156	eci: Invalid parameter size
157	cavv: Required parameter missing
158	capture: Type not allowed for this transaction
159	userAgent: Invalid parameter size
160	urls: Required parameter missing (kind)
161	urls: Invalid parameter format
167	Invalid request JSON
169	Invalid Content-Type
171	Operation not allowed for this transaction
173	Authorization expired
176	urls: Required parameter missing (url)
370	Request failed. Contact Rede
898	PV with invalid ip origin
899	Unsuccessful. Please contact Rede.
1002	Wallet Id: Invalid Parameter Size.
1003	Wallet Id: Required parameter missing.
1018	MCC Invalid Size.
1019	MCC Parameter Required.
1020	MCC Invalid Format.
1021	PaymentFacilitatorID Invalid Size.

returnCode	returnMessage
1023	PaymentFacilitatorID Invalid Format.
1027	SubMerchant: SubMerchantID Invalid Size.
1030	CitySubMerchant Invalid Size.
1032	SubMerchant: Estate Invalid Size.
1034	CountrySubMerchant Invalid Size.
1036	CepSubMerchant Invalid Size
1038	CnpjSubMerchant Invalid Size
3020	Cryptogram: Invalid parameter size.
3025	Deny Category 01: This card should not be used
3026	Deny Category 02: This card should not be used in this PV
3027	Deny Category 03: No cards must be used in this PV
3028	Wallet Processing Type: Invalid Parameter Missing
3029	Wallet Processing Type: Invalid Parameter Size
3030	Wallet Processing Type: Invalid Parameter Format
3031	Wallet Sender Tax Identification: Invalid Parameter Missing
3032	Wallet Sender Tax Identification: Invalid Parameter Size
3033	Wallet Sender Tax Identification: Invalid Parameter Format
3034	SubMerchant: Tax Identification Number Invalid Size.
3035	SubMerchant: Tax Identification Number Invalid Format.
3052	Wallet Code: Required parameter missing.
3053	Wallet Code: Invalid Parameter format.
3054	Wallet Code: Invalid Parameter size.

returnCode	returnMessage
3055	Wallet Code: Parameter not allowed.
3056	Wallet Id: Parameter not allowed
3064	Sai: Invalid parameter size.
3065	Sai: Invalid parameter format.
3066	Sai: Required parameter missing.
3067	Cryptogram: Required parameter missing
3068	Credential Id: Required parameter missing.
3069	Credential Id: Invalid parameter format.
3070	Credential Id: Invalid parameter size.

If you receive return 370 in a sales requisition (automatic or pre-capture), make a reference query to check the status of your transaction.

If return 78 “Transaction does not exist” occurs, the transaction must be retry.

3DS Returns

Authenticated transactions have specific returns and messages.

returnCode	returnMessage
200	Cardholder successfully authenticated
201	Authentication not required
202	Unauthenticated cardholder
203	Authentication service not registered for the merchant. Please contact Rede
204	Cardholder not registered in the issuer's authentication program
220	Transaction request with authentication received. Redirect URL sent
250	onFailure: Required parameter missing

returnCode	returnMessage
251	onFailure: Invalid parameter format
252	urls: Required parameter missing (url/threeDSecureFailure)
253	urls: Invalid parameter size (url/threeDSecureFailure)
254	urls: Invalid parameter format (url/threeDSecureFailure)
255	urls: Required parameter missing (url/threeDSecureSuccess)
256	urls: Invalid parameter size (url/threeDSecureSuccess)
257	urls: Invalid parameter format (url/threeDSecureSuccess)
258	userAgent: Required parameter missing
259	urls: Required parameter missing
260	urls: Required parameter missing (kind/threeDSecureFailure)
261	urls: Required parameter missing (kind/threeDSecureSuccess)
269	ChallengePreference: Invalid parameter format
3000	ColorDepth: Required parameter missing.
3001	DeviceType3ds: Required parameter missing.
3002	JavaEnabled: Required parameter missing.
3003	Language: Required parameter missing.
3004	TimeZoneOffset: Required parameter missing.
3005	ScreenHeight: Required parameter missing.
3006	ScreenWidth: Required parameter missing.
3007	ColorDepth: Invalid parameter size.
3008	DeviceType3ds: Invalid parameter size.
3009	Language: Invalid parameter size.

returnCode	returnMessage
3010	TimeZoneOffset: Invalid parameter size.
3011	ScreenHeight: Invalid parameter size.
3012	ScreenWidth: Invalid parameter size.
3013	ColorDepth: Invalid parameter format.
3014	DeviceType3ds: Invalid parameter format.
3015	JavaEnabled: Invalid parameter format.
3016	Language: Invalid parameter format.
3017	TimeZoneOffset: Invalid parameter format.
3018	ScreenHeight: Invalid parameter format.
3019	ScreenWidth: Invalid parameter format.

Cancellation returns

Canceled transactions have specific returns and messages.

returnCode	returnMessage
351	Forbidden
353	Transaction not found
354	Transaction with period expired for refund
355	Transaction already canceled.
357	Sum of amount refunds greater than the transaction amount
358	Sum of amount refunds greater than the value processed available for refund
359	Refund successful
360	Refund request has been successful

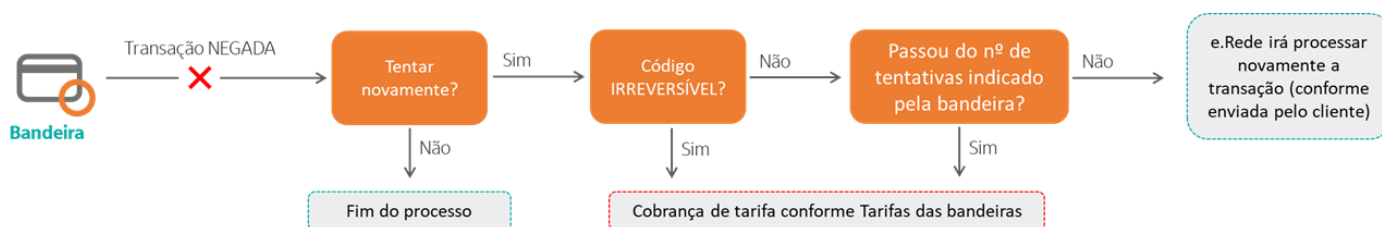
returnCode	returnMessage
362	RefundId not found
363	Callback Url characters exceeded 500
365	Partial refund not available.
368	Unsuccessful. Please try again
369	Refund not found
370	Request failed. Contact Rede
371	Transaction not available for refund. Try again in a few hours
373	No further Refund allowed
374	Refund not allowed. Chargeback requested

Attention: For Code 360, remember that Rede received your cancellation, but you must check it again later to confirm that it was successful.

Brand Fees

Transaction Reattempts

Every time a transaction is denied, the merchant can resubmit that same transaction, reattempting it and trying for an approval. However, card brands established rules for these transaction reattempts which, depending on the return code or how many attempts were made, can result in fees to the merchant. Therefore, to adjust your reattempt systems, the brand rules must be considered.



Brand Rules

Visa Brand

Visa brand uses ABECS codes, separated in 4 categories

VISA	Title	Description	Codes
Category 1	Issuer will never approve.	They inform Merchants/Accreditors that the card was canceled or never existed or that the denial is the result of a permanent restriction or error condition that will prevent future approval.	4, 7, 12, 14, 15, 41, 43, 46, 57, R0, R1, R3
Category 2	Issuer cannot approve at this time.	They Indicate that the negative is the result of a temporary condition such as credit risk, issuer speed controls, or other card restrictions that may allow a retry of the transaction to be approved. In some cases, a denial requires action by the bearer or issuer to remove the restriction before an approval can be obtained.	3, 19, 39, 51, 52, 53, 59, 60, 61, 62, 65, 75, 78, 86, 91, 93, 96, N3, N4, Z5
Category 3	Data quality/review data.	When a data error is identified by the issuer and the transaction is declined as a consequence. Merchants must-revalidate payment data before retrying. Merchants and Accreditors should monitor these denial codes due to potential exposure to fraud.	54, 55, 82, N7, 1A, 70, 6P
Category 4	Generic response codes.	All other Response Codes, many of which are of a technical nature or provide little or no value to Merchants/Accreditors.	All other return codes that are not included in category 1, 2 and 3.

Attention points:

- Code 57 is an exception to the Category 1 rule: Even though it is an irreversible code, it can be retested up to 15 times within a 30-day period.
- Code 14 counts for both category 1 and category 3.

Mastercard brand

Mastercard does not use ABECS codes categorization, instead it uses the complementary return codes known as MAC – Merchant Advice Code, which provide instructions about actions the merchant can take in order to get the transaction approved.

MAC codes accompanies the usual response code when a transaction is denied, and indicates if that transaction can be reattempted or not.

The possible MAC values are:

MAC Value	MAC description	Classification
01	Updated/additional information needed	Reversible
02	Try Again Later	Reversible
03	Do Not Try Again	Irreversible
04	Token requirements not fulfilled for this token type	Reversible
21	Payment Cancellation	Irreversible
24	Retry after 1 hour	Reversible
25	Retry after 24 hours	Reversible
26	Retry after 2 days	Reversible
27	Retry after 4 days	Reversible
28	Retry after 6 days	Reversible
29	Retry after 8 days	Reversible
30	Retry after 10 days	Reversible
40	Non-reloadable prepaid consumer card	Irreversible
41	Consumer single-use virtual card number	Irreversible

MACs 40 and 41 advise the establishment that the card can only be used once.

Mastercard will consolidate some response codes in 3, for exclusive use by Mastercard:

- 79 (Life cycle)
- 82 (Policy)
- 83 (Fraud/Security)

The codes used exclusively by the brand accompanied by the Merchant Advice Code (MAC) function as shown below:

When	Then	And the response code
The issuer declines the transaction using response code 54 (Expired card).	Mastercard will replace code 54 with code 79 (Refusal by life cycle).	Accompanies the proper Merchant Advice Code (MAC).

Attention: MAC codes can accompany any response code, not only Mastercard exclusive codes.

Elo brand

Elo uses ABECS categorization as rule for its reattempt program, the [Brand return code and message standardization table](#) has all the return codes and their respective categorization.

From April 2024, for transactions denied by code 79 – Security/Suspected Fraud, Elo will provide an additional code returned in the field called MAC – Merchant Code Advice, which provides instructions on what actions the merchant can take to approve the transaction.

MAC codes accompany the denial code when a transaction is denied and indicate whether that transaction can be retried (reversible) or not (irreversible).

Possible MAC values are:

Brand return Code (ABECS)	MAC code (merchantAdviceCode)	Description of the combination	Guidance on the retry process
79	1	Need an account update	Do not try again (Irreversible)
79	2	Need to change account information	Retry after correcting data (Reversible)

Hipercard brand

Elo uses ABECS categorization as rule for its reattempt program, the [Brand return code and message standardization table](#) has all the return codes and their respective categorization.

Brand Fees

Visa brand

	Issuer will never approve	Excessive reattempts	Data Quality
	Category code 1	Category 2	Category 3
The maximum limit for retries of denied transactions.	There is no limit, these transactions must not be retried.	15 attempts in 30 days.	10.000 transactions in 30 days.
Fee charged for each attempt that exceeds the limit (domestic transactions).	0.10 USD	0.10 USD	0,10 USD
Fee charged for each attempt that exceeds the limit (international transactions).	0.25 USD	0.25 USD	0,25 USD

Visa brand fees have been in effect since April 2021.

From April 24th, 2023 onward, the Data Quality limit will increase to 25.000 transactions in 30 days on the same PS.

Mastercard brand

	Issuer will never approve	Excessive reattempts
	Any return codes accompanied by MAC 03 or 21	Any transactions, regardless of return code or MAC (Including transactions without MAC)
The maximum limit for retries of denied transactions.	There is no limit, these transactions must not be retried.	7 attempts in 24 hours and limited to 35 attempts in 30 days
Fee charged for each attempt that exceeds the limit (domestic and international transactions).	R\$ 2.50	R\$ 2.00

Mastercard brand fees have been in effect since January 2022.

Important: Excessively retrying transactions that have been denied with a MAC 03 or 21 can result in double billing.

Mastercard double billing example:

Merchant's transaction is denied and is receives a Merchant Advice Code 03 or 21. Then, this merchant retries the same transaction 40 times throughout the month – without surpassing the 7 attempts daily limit – resulting in the following fees:

- R\$ 2.50 x 40 for the Issuer will never approve Fee – Totaling R\$ 100.00

- R\$ 2.00 x 5 (amount of retries exceeding the 35 attempts in 30 days) for the Excessive retries fee – Totaling R\$ 10.00

Thus, the same transaction was billed in both categories, with a total amount of R\$ 110.00.

Elo Brand

	Issuer will never approve	Excessive reattempts
	Irreversible Code	Reversible Code
The maximum limit for retries of denied transactions.	There is no limit, these transactions must not be retried.	15 attempts in 30 days.
Fee charged for each attempt that exceeds the limit (domestic and international transactions).	R\$ 0.80.	R\$ 0.80.

Elo fees have been in effect since August 2022.

Hipercard brand

	Issuer will never approve	Excessive reattempts
	Irreversible Code	Reversible Code
The maximum limit for retries of denied transactions.	There is no limit, these transactions must not be retried.	8 attempts from the first to the last day of the month.
Fee charged when exceeding the limit (domestic and international transactions).	0.03% of the transaction amount per retry, with a minimum of R\$0.15 and a maximum of R\$0.80	R\$1.85 per retry. After exceeding the limit of 8 monthly attempts, the brand will charge for each transaction made after the initial one.

Hipercard fees are in effect since August 2022.

Important: When the limit of 8 possible attempts is surpassed, the brand will charge for each retry made.

Example of Hipercard charging criteria:

Establishment A performs a transaction that is negated by a reversible code. After the denial, over the 30 days of the month, the merchant retries the transaction 15 times, totaling 16 transactions. This will result in a fee of BRL 1.85 x 15, totaling a fee of BRL 27.75.

No use of Zero Dollar

An existing practice to verify a card status is using low amount transaction – validating if there are any restrictions, if the card is valid or if there is enough balance. Once the transaction is approved, it is followed by a reversal. However, this practice – considered inappropriate – will result in fees charged by card brands. To ensure the proper functionality and offer more security for card usage, card brands established rules for account verification transactions.

By using the Zero Dollar function, card validation will be done according to brand specification, with a zero-amount transaction, without any charges to the cardholder.

For more details about Zero Dollar, see [Zero Dollar](#).

Important: Before storing any card data, the use of a Zero Dollar validation is **mandatory**.


Brand rules

Mastercard and brand


Mastercard and Elo established that transactions carried out with amounts of up to R\$ 5,00 and are followed by a reversal will incur in fee charges.

Brand fees

Mastercard Brand

	<p>Transaction with amounts of up to R\$ 5,00 which are followed by a reversal.</p>
<p>Fee charged for every offending transaction</p>	<p>R\$ 0,21</p>

Elo brand

	<p>Transaction with amounts of up to R\$ 5,00 which are followed by a reversal in the same data.</p>
<p>Fee charged for each offending transaction</p>	<p>R\$ 0,2037</p>

Use of Zero Dollar

Brand rules

Visa and Mastercard brands apply a fee for Zero Dollar transactions. This is a cost linked to the use of the Zero Dollar product, that is, to each transaction with zero value sent for approval. Check out the details below:.

Brand fees

Mastercard Brand

Brand	Type	Cost per transaction
Mastercard	Nacional Transactions	R\$ 0,054445
Mastercard	Nacional Transactions	R\$ 0,065334

Visa Brand

Brand	Type	Cost per transaction
Visa	Nacional Transactions	USD 0,0035
Visa	International Transactions	USD 0,025

Preauthorization

Based on a pre-authorization, the merchant can reserve the amount on the customer's card bill, to check their inventory, or send transaction data to their fraud prevention process. After that, and with the analysis completed, the merchant will be able to decide between capturing the transaction or just releasing the reserved limit and returning the pre-captured value of the purchase made on its website.

For the holder, this process is the same as for any purchase on a website, he will see his limit used, and later debit on his invoice in case of confirmation of capture by the merchant. If confirmation is not carried out by the establishment, the cardholder will have access to the reversal information.

Brand rules

For any pre-authorization made for Mastercard transactions, the establishment will be charged for the use of the product.

Brand fees

From September 2023, whenever the merchant uses the pre-authorized product under the Mastercard brand, he will be charged according to the table below:

Mastercard Brand

Type	Applicable to transaction amounts	Cost per transaction
National Transactions	Greater or = R\$ 68,96	0,058% about transaction value
National Transactions	Less than R\$ 68,96	R\$ 0,04
National Transactions	Greater or = R\$ 100,00	0,093% about transaction value
National Transactions	Less than R\$ 100,00	R\$0,04

Authenticated and unauthenticated transactions

Both rates only apply to transactions made with Mastercard® cards. In short, transactions that pass, or fail to pass, on the 3DS will be charged. Basically:

- The amount of 2.9bps (0.029%) will apply to unauthenticated Transactions, with a limit of R\$12,654;
- On Transactions Authenticated via 3DS, the value of 2.9bps (0.029%), with a limit of R\$12,654;
- Regarding Transactions Authenticated via Data Only and via Digital Wallets (Apple, Google and Samsung Pay), none of the fees will be applied.

NOTE: Transactions via Digital Wallets will not incur these fees if they are carried out with all the necessary parameters, indicating a level of security equivalent to an authenticated transaction. It is possible to check the security levels of a transaction through the “ECI” field. Check the details of the values for this field [“here”](#).

To ensure that none of these fees affect your operation, consider implementing these products. Check out more details about Data Only integrations and functionalities [here](#), and Digital Wallets [here](#).

Non-tokenized transactions

Tokenization allows card information to be protected as it is replaced by a token at the time of purchase, ensuring greater security and conversion.

The Visa brand, seeking to encourage the use of tokens, established an additional cost for all transactions that do not use a tokenization method.

To mitigate the incidence of this fee, it is recommended to use tokenization methods, such as:

- [Brand tokenization](#)
- Tokenized and authenticated transactions via [Digital Wallets](#) (Wallets)


Click on each of the items above to find out more.

Brand rules

Visa Brand

All transactions that do not use any tokenization method are subject to fees.

Brand fees

	Non-tokenized transactions
Fee charged for infringing transaction	0.05% of the transaction value

The transfer will start from March 2024.

Staged Digital Wallets

Staged Digital Wallets allow the user to store funds in them before paying for a bill or product.

For the Visa brand, the cost below is expected to start from January 2024:

Brand	Value
Visa	USD 0,01

Cancelation

- For Mastercard debit transactions canceled **from the day following authorization (D+1) up to 10 days after authorization (D+10)**, a fee of R\$10,881 is charged per canceled transaction,
- For Mastercard debit transactions canceled **more than Within 10 days of your authorization**, you will be charged a fee of R\$36.27 per canceled transaction.

Mastercard brand fees have been in effect since October 2023.

Sandbox Tutorial

Starting Point

To start the sandbox integration, it is necessary to access the My Projects menu in the logged area and create a project associated with e.Redde.

At this point, a PV and token will be automatically generated, which should only be used in our sandbox environment. In addition, a postman collection will be made available with examples of requests that will facilitate your integration process. With this information, you are ready to use the services and start submitting integration tests and simulating transactions.

Cards

Our sandbox only works with data from selected cards, as shown in the table below:

Brand	Type	Card	Expire date	Security Code	Token sent from brand (tokenCode)	Token cryptogram sent from brand (tokenCryptogram)
Mastercard	Debit	5277696455399733	jan/35	123	This card does not have a token	This card does not have a token
Mastercard	Credit	5448280000000007	jan/35	123	This card does not have a token	This card does not have a token
Mastercard (BIN 2)	Debit	2223000148400010	jan/35	123	This card does not have a token	This card does not have a token
Mastercard (BIN 2)	Credit	2223020000000005	jan/35	123	This card does not have a token	This card does not have a token
Visa	Debit	4761120000000148	jan/35	123	This card does not have a token	This card does not have a token
Visa	Credit	4235647728025682	jan/35	123	This card does not have a token	This card does not have a token
Hipercard	Credit	6062825624254001	jan/35	123	This card does not have a token	This card does not have a token
Hiper	Credit	6370950847866501	jan/35	123	This card does not have a token	This card does not have a token
Diners	Credit	36490101441625	jan/35	123	This card does not have a token	This card does not have a token
JCB	Credit	3569990012290937	jan/35	123	This card does not have a token	This card does not have a token
JCB (19 dig)	Credit	3572000100200142753	jan/35	123	This card does not have a token	This card does not have a token
Credz	Credit	6367600001405019	jan/35	123	This card does not have a token	This card does not have a token
Elo	Credit	4389351648020055	jan/35	123	This card does not have a token	This card does not have a token
Amex	Credit	371341553758128	jan/35	1234	This card does not have a token	This card does not have a token
Cabal	Credit	6042034400069940	jan/35	123	This card does not have a token	This card does not have a token
Sorocred	Credit	6364142000000122	jan/35	123	This card does not have a token	This card does not have a token

Credsystem	Credit	6280281038975334	jan/35	123	This card does not have a token	This card does not have a token
Banescard	Credit	6031828795629272	jan/35	123	This card does not have a token	This card does not have a token
Visa	Credit	4895370010000005	jan/35	123	483044203527227 9 ou 483044737464978 9 ou 489409371100402 4	AgAAAAAAAIR8CQrXSohbQAAAAAA=
Visa	Debit	4824810010000006	jan/35	123	489409262228016 0 ou 489409602076625 8 ou 489409416734577 0	AAABAKkREQAAAAAAAAAAAAAAAAAA =
Mastercard (BIN 2)	Credit	2223000250000004	jan/35	123	*	ANbuvvxndbK2AAEShHMWGgADFA= =
Mastercard (BIN 2)	Debit	5204970000000007	jan/35	123	*	AOPAIMgflr8UAAIShHMWGgADFA==
Elo	Debit	4514166653413658	jan/35	123	This card does not have a token	This card does not have a token
Elo	Debit	4389356784017450	jan/35	123	This card does not have a token	ANfuuvvxndbK2AAESiXMWGgAEFw==
Elo	Crédito	4389358876174389	jan/35	123	This card does not have a token	Aleb2oot61nRAAHBnZ8HAAADFA==
Elo	Débito	5067230000009011	jan/35	123	This card does not have a token	This card does not have a token

***To process tokenized transactions with the Mastercard card, the card number should be used as the tokenCode. For tokenized transactions with the Visa card, the table's tokenCodes must be used.**

If a transaction is sent with a card other than those informed, the sandbox will return the following error:

Error code	Description
58	Unauthorized. Contact issuer.

Simulate Errors

To simulate error codes, simply send transactions with the values corresponding to the error codes, as shown in the table below:

For the correct functioning of the Elo Brand, it is mandatory to send the securityCode field.

Error code	Amount	Description
53	53	Transaction not allowed for the issuer. Contact Rede.
56	56	Error in reported data. Try again.
57	57	affiliation: Invalid merchant.
58	58	Unauthorized. Contact issuer.
69	69	Transaction not allowed for this product or service.
72	72	Contact issuer.
74	74	Communication failure. Try again.
79	79	Expired card. Transaction cannot be resubmitted. Contact issuer.
80	80	Unauthorized. Contact issuer (Insufficient funds).
83	83	Unauthorized. Contact issuer.
84	84	Unauthorized. Transaction cannot be resubmitted. Contact issuer.
85	85	Please contact issuer.
101	101	Unauthorized. Problems on the card, contact the issuer.
102	102	Unauthorized. Check the situation of the store with the issuer.
103	103	Unauthorized. Please try again.
104	104	Unauthorized. Please try again.
105	105	Unauthorized. Restricted card.
106	106	Error in issuer processing. Please try again.
108	108	Unauthorized. Value not allowed for this type of card.
109	109	Unauthorized. Nonexistent card.
110	110	Unauthorized. Transaction type not allowed for this card.
111	111	Unauthorized. Insufficient funds.
112	112	Unauthorized. Expiry date expired.
113	113	Unauthorized. Identified moderate risk by the issuer.
114	114	Unauthorized. The card does not belong to the payment network.
115	115	Unauthorized. Exceeded the limit of transactions allowed in the period.
116	116	Unauthorized. Please contact the Card Issuer.
117	117	Transaction not found.
118	118	Unauthorized. Card locked.

Error code	Amount	Description
119	119	Unauthorized. Invalid security code.
121	121	Error processing. Please try again.
122	122	Transaction previously sent.
123	123	Unauthorized. Bearer requested the end of the recurrences in the issuer.
124	124	Unauthorized. Contact Rede.
204	204	Cardholder not registered in the issuer's authentication program.
373	373	No further Refund allowed
373	374	Refund not allowed. Chargeback requested
899	899	Unsuccessful. Please contact Rede.
3025	3025	Deny Category 01: This card should not be used
3026	3026	Deny Category 02: This card should not be used in this PV
3027	3027	Deny Category 03: No cards must be used in this PV

Simulate transaction with retroactive date

To simulate a retroactive date transaction, our sandbox is associated with the transaction value.

Transactions sent between **R\$30.01** and **R\$30.99** will return with the capture date retroactive to the current date, where the numbers of the cents are equivalent to the retroactive days.

For example, to simulate the cancellation of a transaction on D-3.

Current date: 10/23

Transaction value: 30.03*(amount: 3003*)

***03 refer to retroactive days**

Generated transaction date: 10/20

Simulate cancellation callback

To simulate the cancellation callback, simply use the cancellation request example available in the sandbox, with the following variations:

- Approved Cancellation: send any value other than 1 in the amount, as long as it meets the rules described in the cancellation documentation section.

- Cancellation Denied: send the value 1 in the amount and the original transaction must have been made with a retroactive date.

Simulate Zero Dollar transaction

To simulate a zero dollar transaction, just send in the request the "amount: 0" and the Zero dollar transaction is performed.

For the correct functioning of the Elo Brand, it is mandatory to send the securityCode field. Remember that in productive transactions the securityCode field is mandatory for Mastercard, Visa and Elo.

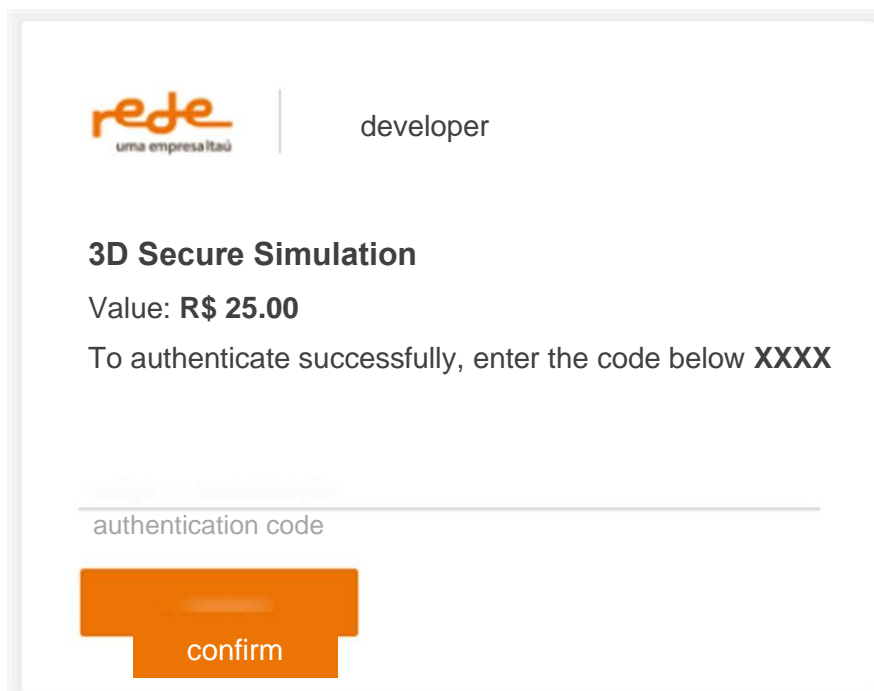
Return Code	Amount	returnMessage	How to simulate
170	0	Zero dollar transaction not allowed for this card.	Use the indicated amount together with any debit card (except ELO cards) from the Test Cards table.
172	0	CVC2 required for Zero Dollar Transaction.	Use the indicated amount together with any Elo card from the Test Cards table without submitting the securityCode.
174	0	Zero dollar transaction success.	Use the indicated amount together with any credit (every brand) or debit (only Elo brand) card from the Test Cards table.
175	0	Zero dollar transaction denied.	Use the indicated amount together with any credit card from the Test Cards table and send subscription field as "True".

Simulate 3DS 2.0 transaction - MPI Rede

To simulate 3DS transaction with MPI Rede, send the information regarding threeD Secure in the request..

The return will be "**220 - Transaction request with authentication received. Redirect URL sent**" containing the filled URL parameter. Copy and paste the URL into your browser to simulate the issuer's authentication screen, as our sandbox environment allows you to emulate the success or failure of the authentication.

To simulate success, enter the code that is displayed on the screen, if you enter a different value, the transaction will be rejected.



rede
uma empresa itaú

developer

3D Secure Simulation

Value: R\$ 25.00

To authenticate successfully, enter the code below **XXXX**

authentication code

confirm

Attention: The screen is a simulation. In production, the screen and information requested vary by issuer.

Another way to stress scenarios and observe the behavior of a transaction via 3DS2.0 in the Sandbox is through the values entered in the “Amount” parameter. Check out the possibilities below:

207 - Authenticated 3DS WITHOUT Challenge (Frictionless Journey)

208 - 3DS authenticated with MANUAL challenge (Manual interaction required by entering challenge code for authentication)

209 - 3DS authenticated with AUTOMATIC challenge (Challenge completed automatically and forwarded to success screen)

NOTE: It is recommended to use the test cards already available in the documentation: [Cards](#)

Simulate transactions with brandTid

As mentioned in the [Recurrence and Card-on-file](#) section, Visa validates the content of the brandTid field, and may deny the transaction if an invalid value is sent.

Currently, the denial given by the Rede for this case is contained in code 58 - Unauthorized. Contact issuer.

To simulate this scenario, send your request with the fields subscription= “true”, storagecard=2 and amount=5.12, with any brandTid value within the field.

Simulate Returns

Simulate MAC

For Mastercard Transactions

- Use a Mastercard card available in our list of cards, linking to it the months and years for each expected return. Example: when sending the card, month: 01; year: 2028 will return MAC 01
- Be receiving ABECS returns. To do this, just make the adjustment in the custom header field, “Transaction-Response”, with the value “brand-return-opened” filled in.

Just simulate the transactions with the values corresponding to the error codes, as shown in the table below:

Month	Year	amount	MAC returned
1	2028	No specific value	01
2	2028	No specific value	02
3	2028	No specific value	03
4	2028	No specific value	04
9	2029	No specific value	21
1	2030	1051	24
2	2030	1051	25
3	2030	1051	26
4	2030	1051	27
5	2030	1051	28
6	2030	1051	29
7	2030	1051	30
5	2031	No specific value	40
6	2031	No specific value	41

Attention: For code 51 exclusive MACs (MACs 24, 25, 26, 27, 28, 29 and 30), the amount “1051” must be sent.

For Elo transactions:

- Use a Elo card available in our list of cards, linking to it the months and years for each expected return. Example: when sending the card, month: 01; year: 2028 will return MAC 01

- Be receiving ABECS returns. To do this, just make the adjustment in the custom header field, “Transaction-Response”, with the value “brand-return-opened” filled in.

Just simulate the transactions with the values corresponding to the error codes, as shown in the table below:

Month	Year	amount	MAC returned
1	2028	No specific value	01
2	2028	No specific value	02

Simulate reattempt returns

- Be receiving ABECS returns. To do this, just make the adjustment in the custom header field, “Transaction-Response”, with the value “brand-return-opened” filled in.

Simulate transactions using any card, with the amount corresponding to the return code, as shown below:

Amount	MAC returned
2001	N01
2002	N02
2003	N03
2004	N04
2099	N99

Security Advice

What is a robot attack?

When fraudsters use payment pages or leaked tokens from APIs to test cards generated by malicious software.

How does it happen?

The robot attack can happen in many ways, the main ones are:

Through the Payment Page:

Sites and platforms which have no security mechanisms to prevent request spams are targeted by fraudsters.

These fraudsters study the web application through the HTTP POST payloads sent from the browser to the site. After learning what is the request which sends card data, they use a script responsible for altering the card data in the payload and automating the request process.

Advice for preventing this type of attack:

1. **Use Google reCAPTCHA or CloudFlare Captcha:** Captcha analyses the traffic and block requests made by bots (robots, automated scripts, etc.) It is important for the Captcha to be linked to the back-end, making it impossible for the fraudster to send new requests. It is important to set up the Captcha's token expiration time, making it short and invalidating it after the request has been sent. Without these settings the attacks can return rapidly.
2. **Validate e-mail or phone number:** Before sending a request to the e.Redes API, a single use code can be sent to the registered e-mail or phone number, in order to validate the transaction. It is important that the validation is linked to the back-end, making it impossible for the fraudster to send requests directly to it. This step must happen prior to the submission of the data to the Payment API.
3. **Block proxies:** Carry out a proxy blockage.
4. **Block temporary e-mail services:** Examples like temp-mail.org, mohmal.com, tempmail.com, etc.

Through compromised tokens:

This happens through vulnerabilities in sites or platforms. Through vulnerabilities the fraudsters can access the databases and obtain the API tokens.

Advice for preventing Integration Key leaking:

1. The key/token can be encrypted using methods more complex than MD5.

Example:

Integration Key stored in the database: vizy6313fx0q0s57xullkyw589a109wd

Integration Key encrypted in SHA256:

9a48909b9f83b827a2c4eec2a340d3534e2ea4a5fb8b0e8abd8eace00872f431

Integration Key encrypted in SHA256 with the addition of salt* (8H1n@5) in the hash generation:

a2c8d68ea678213e0e9471bf10fb888ba8b3f0ad92d869226dbbc7bc68ad8121

*The salt is used to prevent identical passwords from producing identical *hashes*, making the cryptography more secure.

That way, even if a fraudster gets access to the database, the information will be encrypted, making it impossible to guess through brute force attacks.

2. After inserting the Token in the e-commerce platform's management panel (in case it is used), it must be encrypted, making it impossible to be exhibited after being inserted.

Appendix 1 - MCCs allowed in Bill Payment operations via Staged Digital Wallets

Visa

Mastercard

MCC	Nome
4814	Telecommunication services
4899	Cable, Satellite, and Other Paid Television/Radio/Streaming Services
4900	Utilities – Electricity, Gas, Water and Sanitation
6300	Insurance Sales, Underwriting and Premiums
6513	Real Estate Agents And Administrators – Rentals
8211	Elementary and Secondary Schools
8220	Colleges, Universities, Professional and Junior Schools
8241	Correspondence Schools
8244	Business and Secretarial Schools
8249	Commercial and Vocational Schools
8299	Schools and Educational Services [Not Classified Elsewhere]
9311	Tax payment
6012	Financial Institutions—Goods, Services, and Debt Payment
6051	Non-Financial Institutions—Foreign Currency, Non-Fiat Currency (For Example: Cryptocurrency), Money Orders [Not Money Transfers], Account Funding (Not Loading Stored Value), Traveler's Checks, and Debt Payments)
8011	Doctors [Not Elsewhere Classified]
8062	Hospitals
8099	Medical and Health Professional Services [Not Elsewhere Classified]
8111	Legal Services and Attorneys
MCC	Nome
6051	General Mcc - Boleto Payments / Crypto Currencies
4816	Computer Network-Information Services
8299	Schools + Educational Svc-Not Elsewhere Classified
6300	Insurance Sales Underwriting And Premiums
4812	Telecommunication Equipment Incl Telephone Sales
8021	Dentists Orthodontists
8099	Health Practitioners Medical Srvcs-Not Elsewhere
7997	Clubs-Cntry Mbrship(Athlet Rec Sprts Private Golf
8043	Opticians Optical Goods + Eyeglasses
4899	Cable Satellite Other Pay Television Radio Svcs
5983	Fuel Dealers-Coal Fuel Oil Liq Petroleum Wood
7372	Comp Programing Data Prcsng Intgrtd Sys Dsgn Svcs
4900	Utltls-Elctrc Gas Heating Oil Sanitary Water
4814	Telecom Incl Prepaid-Recurring Phone Svcs
7399	Business Services-Not Elsewhere Classified
8011	Doctors (Not Elsewhere Classified)

8220	Colleges Univ Pro Schools Junior Colleges
7392	Consulting Management And Public Relations Svcs
7922	Theatrical Producers(Excl Motion Pix) Ticket Agency
8071	Medical And Dental Laboratories
8062	Hospitals
9311	Tax Payments
8211	Schools Elementary And Secondary
7991	Tourist Attractions And Exhibits
6513	Real Estate Agents And Managers-Rentals
7999	Recreation Services (Not Elsewhere Classified)
5047	Dental-Lab-Med-Ophthalmic Hosp Equip + Supplies
7832	Motion Picture Theaters
8699	Condominium - Houses
7996	Amusement Parks Carnivals Circus Fortune Tellers
5045	Computers Computer Peripheral Equipment Software
5976	Orthopedic Goods-Artificial Limb Stores
8244	Schools Business And Secretarial
8911	Architectural Engineering And Surveying Services
8249	Schools Trade And Vocational
7994	Video Game Arcades-Establishments
8111	Attorneys Legal Services
7941	Athltic Fields Commrc'l Sprt Sprt Clbs Sprt Promotr
9222	Fines
8241	Schools Correspondence
7929	Bands Orchestras + Misc Entrtnrs-Not Elswhr Clas
5975	Hearing Aids-Sales Service Supply Stores
6211	Securities-Brokers-Dealers
8931	Accounting Auditing And Bookkeeping Services
8049	Chiropodists Podiatrists
7911	Dance Halls Schools And Studios
8042	Optometrists Ophthalmologists
7933	Bowling Alleys
8050	Nursing And Personal Care Facilities
7998	Aquariums Dolphinariums And Seaquariums
8041	Chiropractors
7033	Campgrounds And Trailer Parks
7841	Video Entertainment Rental Stores
8031	Osteopathic Physicians
7276	Tax Preparation Service
7932	Pool And Billiard Establishments

7032	Recreational And Sporting Camps
7375	Darren
7993	Video Amusement Game Supplies
7992	Golf Courses-Public
4821	Telegraph Services
7012	Timeshares
4815	Telefonica- Equipment Sales
1477	Colleges Universities And Professional Schools
1384	General Optica
2310	Insurance Sales (C.A.T.)
6310	Insurance Sales
2360	Serviticket (C.A.T.)
2946	Video Tapes Rental Stores (C.A.T.)
7015	Timeshares
1345	Vodafone (Franchises)- Equipment Sales
1455	Tv Rental
1481	Telecommunications Services. Telephonic Traffic
1502	Orange-Mobile Top-Up
1503	Movistar-Mobile Top-Up
2481	Mobile Top-Up (Cat)
2501	Vodafone- Telephone Cards (C.A.T.)
2502	Orange-Mobile Top-Up (Cat)
2503	Movistar- Telephone Cards (C.A.T.)
2732	Cable And Other Pay Television Service
4001	Fuel Station
4811	Telefonica- Services
4817	Movistar- Equipment Sales
4818	Vodafone- Equipment Sales
4819	Amena- Equipment Sales

Amex

MCC	Nome
4814	Telecommunication services
4899	Cable, Satellite, and Other Paid Television/Radio/Streaming Services
4900	Utilities – Electricity, Gas, Water and Sanitation
6300	Insurance Sales, Underwriting and Premiums
6513	Real Estate Agents And Administrators – Rentals

7523	Parking lots and garages
7911	Dance Halls, Studios and Schools
7997	Members Clubs [Sports, Recreation, Athletic], Country Clubs and Private Golf Courses
8011	Doctors [Not Elsewhere Classified]
8062	Hospitals
8099	Medical and Health Professional Services [Not Elsewhere Classified]
8211	Elementary and Secondary Schools
8220	Colleges, Universities, Professional and Junior Schools
8241	Correspondence Schools
8244	Business and Secretarial Schools
8249	Commercial and Vocational Schools
8299	Schools and Educational Services [Not Classified Elsewhere
8351	Child Care Service
9211	Legal Costs, Including Child Support and Alimony
9222	Fitness
9311	Tax payment
9399	Government Services [Not Elsewhere Classified]