

Introduction

Welcome

Hello! Welcome to the e.Redre developer portal.

Here you have access to our test environment (playground) and you can get to know the payment flux in a safe and uncomplicated place.

Our sandbox is similar to the API features in the production environment and you're able to perform integration tests, without being charged for it.

All test logs are stored with the results, creating a history control.

Creating your account

Creating your account is simple, after a few steps you get complete access to our playground enabling you to start the integration tests.

Setting your playground

After creating your account, you must register your “test store” informing the name and de product that will be tested.

Afterwards, it'll be automatically generated a PV and a token that have to be used only in our sandbox environment.

Once all set, you are ready to use our playground and start sending integration tests and emulating transactions.

Modules

Check out the modules to the main open source platforms

Magento



[Learn More](#)

Introduction

Download: [Click here](#) to have access to the Magento module.

The e.Redre module for Magento provides the main credit card brands in a transparent solution without the need for redirection.

Requirements

- Magento Community 1.7.x à 1.9.x
- PHP 5.6
- SPL
- cURL
- DOM

Installation

1. Data backup

For a matter of good practices, backup the store and database before performing any kind of installation.

2. Unable compilation

Before installing any module, disable the compilation option on the back office of the platform:

Log in the administrative panel;

Navigate to System > Tools > Compilation;

3. Installing the e.Rede module

To download the file, follow the instructions below:

- Unzip the file content inside the installation root from the Magento store, mixing with the already existing files;
- On the administrative area of the store, go to **System > Cache Management > Name and Address Options** and change the option **Show Tax/VAT Number** to **Required**.

OpenCart



[Learn More](#)

Introduction

Download: [Click here](#) to have access to the OpenCart module.

The e.Rede module for OpenCart provides the main credit card brands in a transparent solution without the need for redirection.

Requirements

- Opencart 1.5.4.x à 1.5.6.x
- VQmod

- PHP 5.6
- SPL
- cURL
- DOM

Installation

1. Data backup

For a matter of good practices, backup the store and database before performing any kind of installation.

2. Installing the e.Redre module

After downloading the file, follow the instructions below:

- Unzip the file content inside the installation root of the Opencart store, mixing the already existing files;
- On the administrative panel of the store, go to Extensions > Payments, locate the e.Redre module and click on Install.
- Run the file "index.php" once, accessing: Ex.: <https://yourstore.com.br/rede/install>

This command will create the files and configuration required for the e.Redre module operation in your store.

Woo commerce



Learn More

Introduction

Download: [Click here](#) to have access to the WooCommerce module.

The e.Redre module for WooCommerce provides the main credit card brands in a transparent solution without the need for redirection.

Requirements

- Woocommerce 2.6.x
- PHP 5.6
- SPL
- cURL
- DOM

Installation

1. Data backup

For a matter of good practices, backup the store and database before performing any kind of installation.

2. Installing the e.Red module

After downloading the file, follow the instructions below:

- Unzip the package content inside the folder wp-content > plugins found on the WordPress installation;
- On the administrative panel of the store, go to Plugins, locate the e.Red WooCommerce module and click on Activate.
- Then, on the Woocommerce menu, go to Configurations > Finish purchase and locate the e.Red module inside the payment options.

Prestashop



[Learn More](#)

Introduction

Download: [Click here](#) to have access to the Prestashop module.

The e.Red module for Prestashop provides the main credit card brands in a transparent solution without the need for redirection.

Requirements

- Prestashop 1.6.x
- PHP 5.6
- SPL
- cURL
- DOM

Installation

1. Data backup

For a matter of good practices, backup the store and database before performing any kind of installation.

2. Installing the e.Red module

After downloading the file, follow the instructions below:

- Unzip the package file inside the folder modules, located on the installing root of the Prestashop store;

- On the administrative panel of the store, go to Modules > Modules and Services, locate the e.Redex module and click in Install.
-

Libraries

Libraries

Check out the SDKs available for the e.Redex

PHP

Introduction

Download: [Click here](#) to have access to the PHP SDK.

The e.Redex SDK in PHP provides full support to the integration of the payment ways from Redex in e-commerce stores and platforms. It is possible to create, process and manage the payments with it.

You can have a more comprehensive idea of our API, by accessing the documentation.

Requirements

- [php](#), version 5.6
- [curl](#)
- [json](#)
- [mbstring](#)

Installation

To use the SDK, it is required to copy your content (e.redex folder) to a directory of your application and use the SDK class loader.

```
require_once({SDK's location} + '/eredex/Classloader.php')
```

Configuration

The SDK configuration is made dynamically when creating the object "Acquirer". It is required to inform the following parameters:

- Affiliation: The PV number
- Token: The token linked to the PV
- Environment: The environment ("Sandbox" or "Production") of the api that will be used.

Example

Inform your credentials:

```
$acquirer = new Acquirer("123", "abc123", EnvironmentType::Production)
```

Credit authorization

```
$request = new TransactionRequest();
$request->setCapture("false");
$request->setAmount(1000);
$request->setReference("123456X");
$request->setCardNumber("123456");
$request->setSecurityCode("123");
$request->setExpirationMonth(6);
$request->setExpirationYear(19);
$request->setCardHolderName("NOME PORTADOR");
$request->setInstallments(1);

$response = $acquirer->authorize($request);
```

Capture

```
$tid = "10987654321";
$request = new TransactionRequest();
$request->setAmount(1000);

$response = $acquirer->capture($tid, $request);
```

Cancellation

```
$tid = "10987654321";
$request = new RefundRequest();
$request->setAmount(1000);

$response = $acquirer->refund($tid, $request);
```

Query

```
$tid = "10987654321";
$query = new Query("123", "abc123", EnvironmentType::Production);

$response = $query->getTransactionByTid($tid);
```

License

Apache License 2.0

JAVA

Download: [Click here](#) to have access to the Java SDK.

The e.Redes SDK in Java provides full support to the integration of the payment ways from Rede in e-commerce stores and platforms. It is possible to create, process, and manage the payments with it.

You can have a more comprehensive idea of our API, by accessing the e.Redes's documentation.

Requirements

- [Java](#), version 1.6 or above
- [Maven](#)

Installation

Access the SDK directory where the file pom.xml is and run the command:

```
mvn package
```

Extract the generated jar and import it on your project.

Configuration

The SDK configuration is made dynamically when creating an object "Acquirer". It is required to inform the following parameters: - Affiliation: The PV number – Token: The token linked to the PV – Environment: The environment ("Sandbox" or "Production") of the api that will be used.

Example

Inform your credentials:

```
Acquirer acquirer = new Acquirer("123", "abc123", EnvironmentType.PRODUCTION);
```

Credit authorization

```
TransactionRequest request = new TransactionRequest();
request.setCapture(false);
request.setReference("pedido01");
request.setAmount(1000);
request.setCardHolderName("Nome Portador");
request.setCardNumber("123");
request.setExpirationMonth(12);
request.setExpirationYear(18);
request.setSecurityCode("123");
request.setInstallments(1);

TransactionResponse response = acquirer.authorize(request);
```

Capture

```
String tid = "10987654321";
TransactionRequest request = new TransactionRequest();
request.setAmount(1000);

TransactionResponse response = acquirer.capture(request, tid);
```

Cancellation

```
String tid = "10987654321";
RefundRequest request = new RefundRequest();
request.setAmount(1000);
```

```
RefundResponse response = acquirer.refund(tid, request);
```

Query

```
Query query = new Query("123", "abc123", EnvironmentType.PRODUCTION);  
String tid = "10987654321";  
  
QueryTransactionResponse response = query.getTransactionByTid(tid);
```

License

Apache License 2.0

RUBY

Download: [Click here](#) to have access to the Ruby SDK.

The e.Redre SDK in Ruby provides full support to the integration of the payment ways from Rede in e-commerce stores and platforms. It is possible to create, process, and manage the payments with it.

You can have a more comprehensive idea of our API, by accessing the e.Redre's documentation.

Requirements

- [Ruby](#)

Installation

Generate the gem from the SDK fonts, run the command:

```
gem build erede/acquirer.gemspec
```

Install the gem using the following command:

```
gem install erede
```

Configuration

The SDK configuration is made dynamically when creating an object "Acquirer". It is required to inform the following parameters: - Affiliation: The PV number – Token: The token linked to the PV – Environment: The environment ("Sandbox" or "Production") of the api that will be used.

Example

Inform your credentials:

```
acquirer = Rede::Acquirer.new("123", "abc123", Rede::EnvironmentType::PRODUCTION)
```

Credit authorization

```
request = Rede::TransactionRequest.new()  
request.capture = false  
request.card_number = "1234"  
request.security_code = "123"
```



```
request.expiration_month = "12"  
request.expiration_year = "2023"  
request.amount = 1000  
request.card_holder_name = "NOME PORTADOR"  
request.installments = 1  
request.reference = "pedido01"  
  
response = acquirer.authorize(request)
```

Capture

```
tid = "10987654321"  
request = Rede::TransactionRequest.new()  
request.amount = 1000  
  
response = acquirer.capture(tid, request)
```

Cancellation

```
tid = "10987654321"  
request = Rede::RefundRequest.new()  
request.amount = 1000  
  
response = acquirer.refund(tid, request)
```

Query

```
tid = "10987654321"  
query = Rede::Query.new("123", "abc123", Rede::EnvironmentType::PRODUCTION)  
  
response = query.get_transaction_by_tid(tid, request)
```

License

Apache License 2.0

.NET

Download: [Click here](#) to have access to the .NET SDK.

The e.Redes SDK in .NET provides full support to the integration of the payment ways from Rede in e-commerce stores and platforms. It is possible to create, process, and manage the payments with it.

You can have a more comprehensive idea of our API, by accessing the e.Redes's documentation.

Requirements

- [.NET Framework 4.5](#)

Installation

On the SDK directory, perform the build of the project Rede.Acquiring.SDK.Rest. In your project, it is required to add references to all the DLLs generated by the build.

Configuration

The SDK configuration is made dynamically when creating an object "Acquirer". It is required to inform the following parameters:

- Affiliation: The PV number
- Token: The token linked to the PV
- Environment: The environment ("Sandbox" or "Production") of the api that will be used.

Example

Inform your credentials:

```
var acquirer = new Acquirer("123", "abc123", EnvironmentType.Production);
```

Credit Authorization

```
var request = new TransactionRequest();
request.Capture = false;
request.Amount = 1000;
request.CardHolderName = "Nome Portador";
request.CardNumber = "1234";
request.ExpirationMonth = 8;
request.ExpirationYear = 19;
request.SecurityCode = "123";
request.Installments = 1;
request.Reference = "pedido01";

TransactionResponse response = acquirer.Authorize (request);
```

Capture

```
var tid = "10987654321";
var request = new TransactionRequest();
request.Amount = 1000;

TransactionResponse response = acquirer.Capture(tid, request);
```

Cancellation

```
var tid = "10987654321";
var request = new RefundRequest();
request.Amount = 1000;

RefundResponse response = acquirer.Refund(tid, request);
```

Query

```
var tid = "10987654321";
var query = new Query("123", "abc123", EnvironmentType.Production);

QueryTransactionResponse response = query.GetTransactionByTid(tid);
```

License

Apache License 2.0

Python

Download: [Click here](#) to have access to the Python SDK.

The e.Redre SDK in Python provides full support to the integration of the payment ways from Rede in e-commerce stores and platforms. It is possible to create, process, and manage the payments with it.

You can have a more comprehensive idea of our API, by accessing the e.Redre's documentation.

Requirements

- [Python](#)
- [pip](#)

Installation

To perform the SDK installation on the development environment, run the following command on the root directory of the SDK:

```
pip install --e .
```

Configuration

The SDK configuration is made dynamically when creating an object "Acquirer". It is required to inform the following parameters:

- Affiliation: The PV number
- Token: The token linked to the PV
- Environment: The environment ("Sandbox" or "Production") of the api that will be used.

Example

Inform your credentials:

```
acquirer = Acquirer("123", "abc123", EnvironmentType.production);
```

Credit authorization

```
request = TransactionRequest ()
request.capture = False
request.reference = "pedido01"
request.amount = 1000
request.installments = 1
request.cardHolderName = "Nome Portador"
request.cardNumber = "1234"
request.expirationMonth = 8
```

```
request.expirationYear = 21  
request.securityCode = "123"
```

```
response = acquirer.authorize (request)
```

Capture

```
tid = "10987654321"  
request = TransactionRequest()  
request.amount = 1000
```

```
response = acquirer.capture(tid, request)
```

Cancellation

```
tid = "10987654321"  
request = RefundRequest ()  
request.amount = 1000
```

```
response = acquirer.refund(tid, request)
```

Query

```
tid = "10987654321"  
query = Query("123", "abc123", EnvironmentType.production);
```

```
response = query.query_transaction_by_tid(tid)
```

License

Apache License 2.0

NodeJs

Download: [Click here](#) to have access to the NodeJS SDK.

The e.Redes SDK in NodeJS provides full support to the integration of the payment ways from Rede in e-commerce stores and platforms. It is possible to create, process, and manage the payments with it.

You can have a more comprehensive idea of our API, by accessing the e.Redes's documentation.

Requirements

- [npm](#)

Installation

To add the SDK to your module/project, it is required to run the following command:

```
npm install {diretório da SDK}/eredes
```

Configuration

The SDK configuration is made dynamically when creating an object “Acquirer”. It is required to inform the following parameters: - Affiliation: The PV number – Token: The token linked to the PV – Environment: The environment (“Sandbox or Production”) of the api that will be used.

Example

Inform your credentials:

```
var erede = require('erede');  
var acquirer = new erede.Acquirer('123', 'abc123', erede.EnvironmentType.PRODUCTION);
```

Credit authorization

```
var request = new erede.TransactionRequest();  
request.setCapture(false);  
request.setReference('pedido01');  
request.setAmount(1000);  
request.setInstallments(1);  
request.setCardHolderName('Nome Portador');  
request.setCardNumber('1234');  
request.setExpirationMonth(12);  
request.setExpirationYear(21);  
request.setSecurityCode('123');  
  
acquirer.Authorize(request, function(resFail, resSuccess)  
  var response = resSuccess;  
});
```

Capture

```
var tid = '10987654321';  
var request = new erede.TransactionRequest();  
request.setAmount(1000);  
  
acquirer.Capture(tid, request, function(resFail, resSuccess){  
  var response = resSuccess;  
});
```

Cancellation

```
var tid = '10987654321';  
var request = new erede.RefundRequest();  
request.setAmount(1000);  
  
acquirer.Refund(tid, request, function(resFail, resSuccess){  
  var response = resSuccess;  
});
```

Query

```
var tid = '10987654321';  
var query = new erede.Query('123', 'abc123', erede.EnvironmentType.PRODUCTION);
```

```
query.QueryTransactionByTid(tid, function(resFail, resSuccess){  
    var response = resSuccess;  
});
```

License

Apache License 2.0

Documentation

Presentation

The e.Redre is a practical and safe online payment solution to perform sales on the internet with all tranquility.

With different kinds of integration, the solution performs the capture and processing of financial transactions directly by Rede, that is, without needing an intermediary, providing payments with credit and debit cards of the main card brands of the market.

Besides, the e.Redre provides several functionalities to aggregate even more value to your customer's business, with a focus on the conversion of sales and greater managing control.

The purpose of this guide is to help the developer to integrate his application with the e.Redre, listing all functionalities and methods, with examples of messages to be sent and received.

The e.Redre solution was developed thinking on the ease for the establishment that wishes to use the API without having to install new systems. The main advantages of using an API are interoperability between distinct applications and physically distant, portability between platforms, integration ease, costs reduction for data transportation and universal format.

Glossary

To make it easier to understand, we have created a glossary with the most used terms related to e-commerce, acquiring and cards.

- **Authorization:** a process that sensitizes the cardholder's credit limit along with the issuing bank, usually used for prevention analysis, credit limit analysis and data validation of the card used. The authorization doesn't generate a charge on the buyer's invoice.
- **Capture:** a process that confirms an authorized transaction. After the capture, the amount is charged from the cardholder's card, generating a charge on the invoice.
- **Cancellation:** a process that returns to the buyer the authorized amount or captured on the card.
- **PV (filiation number):** identifying code generated by Rede for the filiated establishments. The PV ('Ponto de Venda' in Portuguese) is unique for each establishment.
- **Token:** a security code generated by Rede, used to ensure the transaction integrity. It is part, along with the PV, of the API authentication credentials.
- **Issuer:** is the financial institution that issues the credit or debit card.
- **Holder:** is the card proprietary, product buyer.
- **Establishment:** is the entity responsible for the e-commerce (store or virtual service).

- **TID:** is the unique identifier of a transaction, composed by up to 20 characters generated by Rede. This identifier does not repeat.
- **MPI:** system responsible for authenticating along with the issuer, the debit and credit transactions.
- **SecureCode:** is the MasterCard's protected buying system that certifies along with the issuer, the cardholder, validating data that only him and the bank have. It follows the universal 3D Secure standard.
- **Verified by Visa (VBV):** is the Visa's protected buying system that certifies along with the issuer, the cardholder, validating data that only him and the bank have. It follows the universal 3D Secure standard.

Technical prerequisites

The integration mechanism with e.Redre is simple, in a way that only knowledge in programming languages for web, HTTP/HTTPS requisitions, and JSON file manipulation, are required to successfully implement the solution.

Accreditation

To request the e.Redre accreditation and perform the integration to your application, contact the Rede Call Center:

4004 4490 (capitals and metropolitan regions)

0800 728 4490 (other locations)

When the accreditation is made, the person responsible for the establishment will be notified via e-mail with the filiation number (PV), orientations to access the Rede portal and his credentials for integration.

Homologation and SSL certificate

To transact with e.Redre's API, it is required for the establishment to have installed on the payment page, an SSL security certificate with encryption 2048 bits or above, to ensure the secrecy of the transferred information and certify the cardholder that he's really accessing the website desired, avoiding fraud problems.

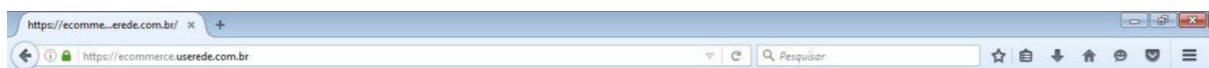
To ensure that the establishments have the SSL certificate installed, Rede does the homologation process automatically of the store or virtual service of the establishment after the performing of the first transaction.

Periodically, the homologation process is performed and rede reserves the right of suspending the use of the platform until the store or virtual service are suitable to the requested security norms.

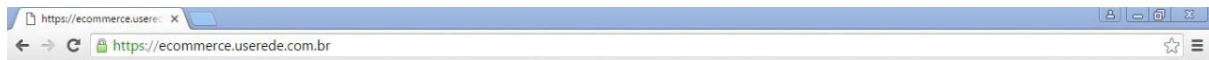
To identify if the page has the SSL certificate, when accessing the website, the URL must be displayed with the "https" protocol, enabling the view of the security padlock on the browsers.

Examples:

Firefox:



Google Chrome:



Internet Explorer



In case the establishment has been suspended for not being certified, access the Rede portal on the menu e.Redre > Configurações > Solicitar homologação and click in “Solicitar homologação” after the SSL certificate regularization.

Filiation and Token

For the establishment to start transacting with e.Redre, it is required to configure the API with your credentials: PV (filiation number) and token.

The token is a confidential key, generated on Rede portal. To generate it, make sure that your user has the administrator profile. Access the menu: e.Redre > Configurations > Token generation and click in "Generate token".

In case of forgotten or lost token, a new one must be generated and the API configuration will have to be updated with the new token. This way the transactions will keep on being sent to Rede.

Credentials

Before starting the transactions, it is required to perform the configurations for the requests on the e.Redre API. For that, make sure that you already have the PV and token correspondent to the environment kind that you want to use (Sandbox or Production).

To generate the test credential, it is required to create your account on the developer portal. On the logged area, create your "test store" informing the name and the product that will be tested. At this moment, it'll be automatically generated an PV and test token that must be used only in our sandbox environment.

The data for authorization must be sent on the header Authorization of the HTTP requisition, following the Basic Authentication of HTTP standard.

The authentication will have to contain the filiation number (PV) and the registered token encrypted on the **base64** format, as on the example below:

Filiation and Token: 00000000:4913bb24a0284954be72c4258e229c99

Base64: MDAwMDAwMDA6NDkxM2JiMjRhMDI4NDk1NGJINzJjNDI1OGUyMjJjOTk=

Once the integration on the sandbox and the homologations tests are made, the following step is to migrate to the production environment. To do so, the filiation number (PV) and token must be replaced by the specific credentials of the establishment, as well the production endpoints.

Endpoints

The endpoints are URLs that will be used for the request of a given service. They may vary, depending on the environment and HTTP method.

The composition is made as follows:

- Base URL

- API version
- Service

Example:

Sandbox	https://api.userede.com.br/desenvolvedores/v1/transactions
Production	https://api.userede.com.br/erede/v1/transactions

For each service type, a complement will be added to the URL so that afterward, the requisition is made. During this documentation, we'll list all of them.

HTTP Methods

The HTTP methods for RESTful services will be frequently used for requisition of transactions.

HTTP VERB	DESCRIPTION
POST	Used for features creation or to send information that will be processed. For example, creation of a transaction.
GET	Used for queries of already existing features. For example, transactions query.
PUT	Used for the authorization of an already existing feature. For example, the capture of a transaction previously authorized.

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

Formatting

Encoding

To perform transactions using the e.Redre API, it'll be required to configure it in your application, the use of encoding UTF-8.

JSON

JSON (JavaScript Object Notation) is a standard for data description for interchange between systems. It is simpler and lighter than XML. By default, the entire API transits JSON, both to receive information (POST and PUT methods) and return (GET method).

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

Fields of Datetime type

All the attributes of the Datetime type, both attributes that are returned in objects and those that are passed as parameters on the 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

There are two ways to do a transaction. The establishment may choose between a posterior capture or automatic capture.

The authorization with posterior capture sensitizes the cardholder's limit with the amount of the transaction, however, doesn't generate charges on the invoice as long as the establishment doesn't make a confirmation (capture).

On the other hand, on the authorization with automatic capture, the amount of the transaction is confirmed instantly, not requiring a capture transaction (confirmation).

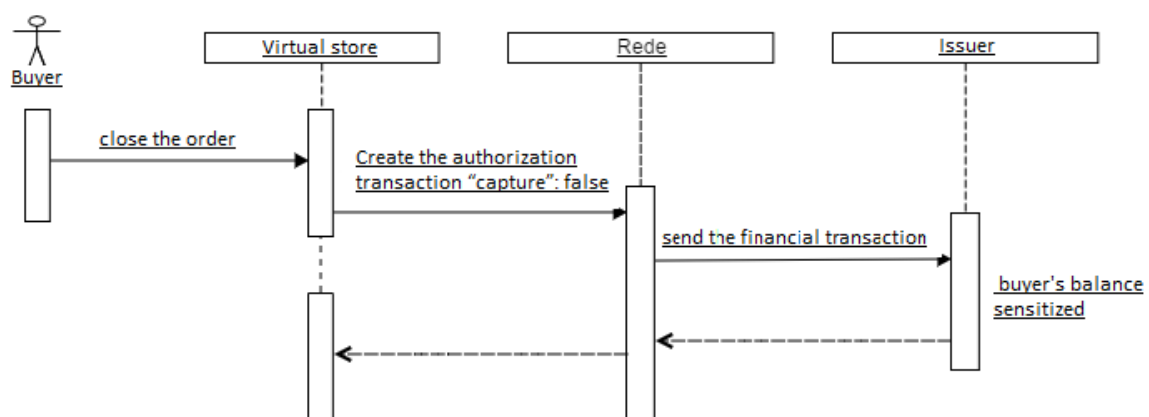
Authorization

The authorization is the first step to perform a transaction. The amount of the transaction sensitizes the cardholder's limit, however, doesn't generate charges on the invoice as long as the establishment doesn't make a confirmation (capture).

If the transaction is not captured before the deadline, it is automatically canceled. The deadline vary according to the field of business of the establishment.

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

The diagram below shows the authorization transaction flux without the automatic capture:



POST: <https://api.userede.com.br/erede/v1/transactions>

Request:

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
capture		Boolean	No	Defines if the transaction will have automatic or posterior capture. Not sending this field will be considered automatic capture (true).

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
kind		Alphanumeric	No	<p>Kind of transaction to be performed.</p> <ul style="list-style-type: none"> For credit transactions, use credit For debit transactions, use debit <p>Not sending 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>The total amount of the transaction without thousands separator and decimal separator. Examples: R\$ 10,00 = 1000 R\$ 0,50 = 50</p>
installments	Up to 2	Numeric	No	<p>The number of installments in which the transaction will be authorized. From 2 to 12. Not sending this field, it will be considered the value "1".</p>
cardHolderName	Up to 30	Alphanumeric	No	Cardholder name printed on the card.
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 Ex.: 2021 or 21</p>
securityCode	Up to 4	Alphanumeric	No	<p>Card security code (usually located on the card back) To this parameter ensures a greater possibility of transaction approval.</p>
softDescriptor	Up to 13	Alphanumeric	No	A customized phrase that will be printed on the cardholder's invoice.

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
subscription		Boolean	No	<p>Informes the issuer whether the transaction comes from a recurrence. If the transaction is a recurrence, send true. Otherwise, send false.</p> <p>Not sending this field will be considered the value false.</p> <p>Rede does not manage the recurrence schedule, it only allows the retailers to indicate whether the originated transaction is from a recurrent plan.</p>
Origin	Up to 2	Numeric	No	<p>Indicates the transaction origin.</p> <ul style="list-style-type: none"> e.Redes - 1 VISA Checkout - 4 MasterPass - 6 <p>Not sending this field will be considered an e.Redes transaction (Origin = 1).</p>
distributorAffiliation	Up to 9	Numeric	No	Distributor affiliation number (PV)

Response:

NAME	SIZE	TYPE	DESCRIPTION
returnCode	Up to 3	Alphanumeric	Transaction return code
returnMessage	Up to 256	Alphanumeric	Transaction return message
reference	Up to 16	Alphanumeric	Order number generated by the establishment
tid	Up to 20	Alphanumeric	Transaction unique identifier number
nsu	Up to 12	Alphanumeric	Sequential number returned by Rede
authorizationCode	6	Alphanumeric	Authorization number of the transaction returned by the card issuer.

NAME	SIZE	TYPE	DESCRIPTION
dateTime		Datetime	Transaction date on the format YYYY-MM-DDThh:mm:ss.sTZD
amount	Up to 10	Numeric	Transaction total amount without the separator of thousands and decimals.
cardBin	6	Alphanumeric	First 6 digits of the card
last4	4	Alphanumeric	Last 4 digits of the card

Sample JSON:

Request:

```
{
  "capture": false,
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2020,
  "securityCode": "235"
}
```

Response:

```
{
  "reference": "pedido123",
  "tid": "8345000363484052380",
  "nsu": "663206341",
  "authorizationCode": "186376",
  "dateTime": "2017-02-28T08:54:00.000-03:00",
  "amount": 2099,
  "cardBin": "544828",
  "last4": "0007",
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://api.userede.com.br/eredev1/transactions/8345000363484052380"
    }
  ],
}
```

```

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

```

Sample JSON

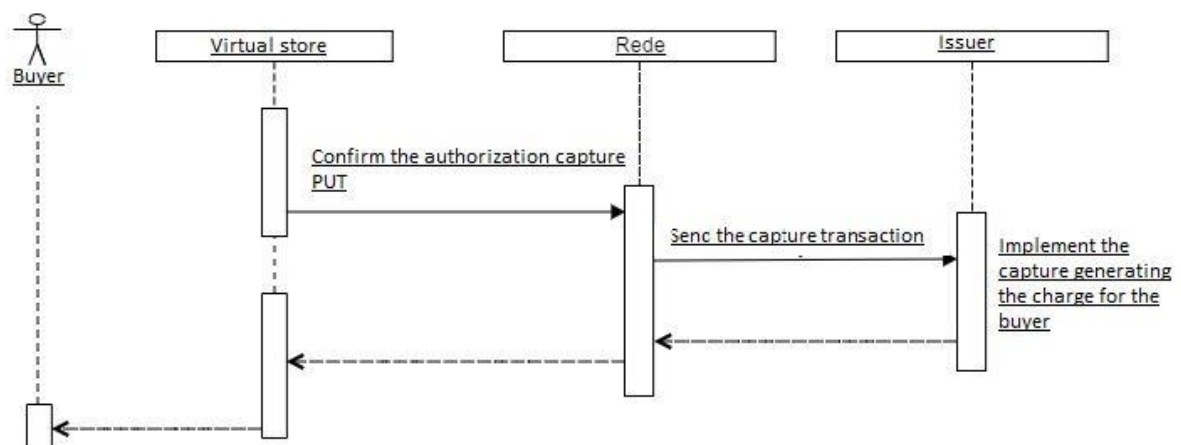
Capture

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

The authorization must have to be captured within the maximum period according to the field of business of the establishment.

IMPORTANT: Always wait for the transaction answer before performing a new capture attempt of it.

The diagram below shows the capture transaction flux:



PUT: [https://api.userede.com.br/erede/v1/transactions/\[tid\]](https://api.userede.com.br/erede/v1/transactions/[tid])

Request:

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
amount	Up to 10	Numeric	No	Capture amount without the separator of thousands and decimals. Examples: R\$ 10,00 = 1000 R\$ 0,50 = 50

Response:

NOME	TAMANHO	TIPO	DESCRIÇÃO
reference	Up to 16	Alphanumeric	Order number generated by the establishment
tid	Up to 20	Alphanumeric	Transaction unique identifier number
nsu	Up to 12	Alphanumeric	Sequential number returned by Rede
authorizationCode	6	Alphanumeric	Authorization number of the transaction returned by the card issuer.
dateTime		Datetime	Transaction date on the format YYYY-MM-DDThh:mm:ss.sTZD
returnCode	Up to 3	Alphanumeric	Transaction return code
returnMessage	Up to 256	Alphanumeric	Transaction return message

Sample JSON:

Request:

PUT: <https://api.userede.com.br/eredev1/transactions/8345000363484052380>

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

Response:

```
{
  "reference": "pedido123",
  "tid": "8345000363484052380",
  "nsu": "7648531",
  "authorizationCode": "186376",
  "dateTime": "2017-02-28T08:54:00.000-03:00",
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
```

```
{
  "method": "GET",
  "rel": "transaction",
  "href": "https://api.userede.com.br/erede/v1/transactions/9274256037511432483"
},
{
  "method": "POST",
  "rel": "refund",
  "href": "https://api.userede.com.br/erede/v1/transactions/9274256037511432483/refunds"
}
]
```

3D Secure

3D Secure authenticated transactions (or 3DS) are transactions that need an additional authentication to ensure a greater safety for the cardholder in online purchases.

The 3DS authentication is made through the validation of data that only the cardholder and the issuer have, for example, card password, birth date, security code, bank token. In case of authentication success, the issuer assumes the financial risk of the transaction (chargeback).

Currently, only Mastercard and Visa card brands have these services active on e.Redde, identified by Mastercard SecureCode and Verified by Visa, respectively.

The 3DS authentication is mandatory for all transactions made with debit cards. For credit cards, the use is optional.

The MPI (merchant plug-in) is the service that provides the integration of the establishment with different issuers, according to the card brands certifications for the processing of the 3DS authentication.

The e.Redde provides two ways to use the 3DS service, through Rede MPI or Client MPI.

- Rede MPI: service already embedded in the e.Redde platform, without needing additional hiring.
- Client MPI: service hired additionally by the establishment for the integration to e.Redde, may be of the credit or debit kind without Rede influence.

To enable 3DS transactions, the issuers must also be prepared to receive the cardholder's authentication information. The main issuers in Brazil already make this service available to their customers.

MPI Rede

The e.Redde has the MPI embedded in its platform. Therefore, to signalize that the MPI used is by Rede, follow the instructions of the *embedded* parameter, contained on the table below.

The transactions that use the 3DS service with the Rede MPI may be of the credit or debit type and must be sent along with the BODY of the authorization transaction.

If a credit transaction can't be successfully authenticated, there's another way of proceeding without the due 3DS authentication. This option makes the transaction risk to belong to the establishment

(as any other type of transactions). For that, follow the instructions of the onFailure parameter on the table below.

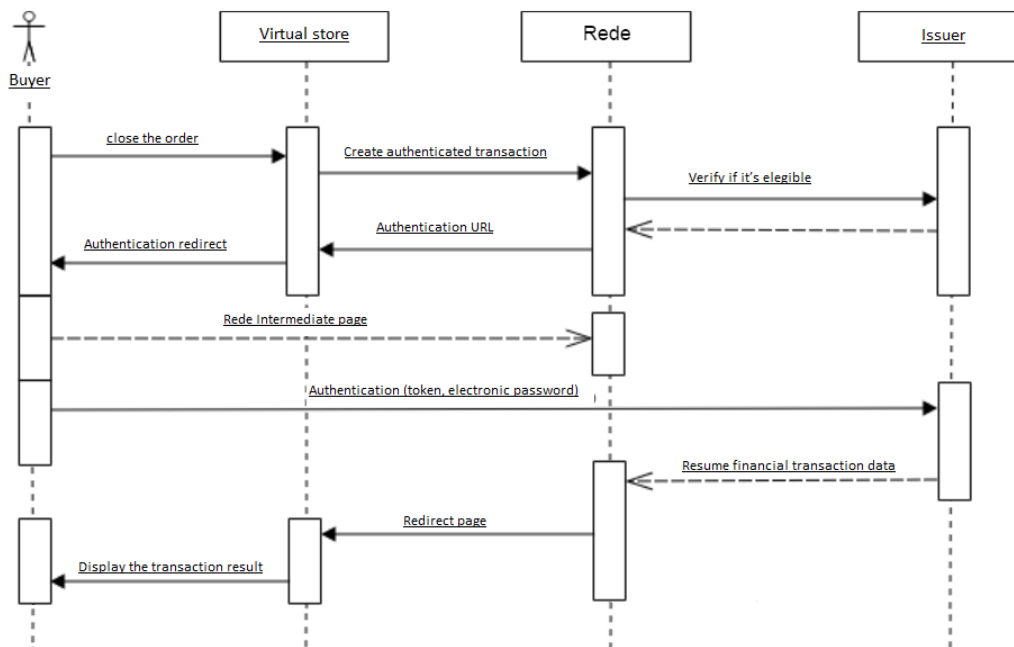
For debit transactions, the amount of this parameter is automatically set due to the mandatory authentication.

To enable the 3DS service, contact Rede through the e-mail ecommerce@userede.com.br and send the following information:

- Company CNPJ (National Register of Legal Entities)
- Trading name
- PV (filiation number)

In 7 business days, Rede will return informing the solicitation status of the 3DS enablement.

The diagram below shows the authenticated transaction process through Rede's MPI:



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

POST: <https://api.userede.com.br/erede/v1/transactions>

Request:

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
threeDSecure		threeDSecure		

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
threeDSecure/embedded		Boolean	No	<p>Informes whether the used MPI service will be from Rede or third parties'.</p> <p>true: uses the MPI service from Rede.</p> <p>false: uses a third party MPI.</p> <p>Not sending of this field will be considered Rede MPI usage.</p>
threeDSecure/onFailure		Alphanumeric	No	<p>Defines how to proceed with the transaction if the 3DS does not succeed.</p> <p>continue: proceeds with the financial transaction, even if the authentication fails.</p> <p>decline: does not proceed with the financial transaction in case the authentication fails.</p> <p>For debit transactions, the value of this parameter is automatically set to "decline" due to the mandatory authentication .</p>
threeDSecure/userAgent	Up to 500	Alphanumeric	Yes	Browser indicator used by the buyer at the moment of the purchase.
urls		urls		
urls/kind		Alphanumeric	Yes	<p>The field that identifies what is the url kind.</p> <ul style="list-style-type: none"> threeDSecureSuccess threeDSecureFailure
urls/url	Up to 87	Alphanumeric	Yes	Field to inform the url that the buyer will have to be redirected after the authentication and notified via postback (application/x-www-form-urlencoded) with the transaction data. Click here for more information.

Response:

NAME	SIZE	TYPE	DESCRIPTION
------	------	------	-------------

NAME	SIZE	TYPE	DESCRIPTION
threeDSecure		threeDSecure	
threeDSecure /embedded		Boolean	Informs whether the used MPI service will be from Rede or third parties'
threeDSecure /url	Up to 500	Alphanumeric	Authentication url returned by the MPI system
threeDSecure /returnCode	3	Alphanumeric	A return code of the transaction with 3DS (check out the table "3DS returns")
threeDSecure /returnMessage	Up to 256	Alphanumeric	Return message of the transaction with 3DS (check out the table "3DS returns")

JSON sample:

Request:

```
{
  "kind": "debit",
  "capture": true,
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "2223000148400010",
  "expirationMonth": 12,
  "expirationYear": 2020,
  "securityCode": "235",
  "threeDSecure": {
    "embedded": true,
    "onFailure": "decline",
    "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"
  },
  "urls": [
    {
      "kind": "threeDSecureSuccess",
      "url": "https://redirecturl.com/3ds/success"
    },
    {
      "kind": "threeDSecureFailure",
      "url": "https://redirecturl.com/3ds/failure"
    }
  ]
}
```

```
}  
]  
}
```

Response:

```
{  
  "dateTime": "2017-02-12T10:54:00.000-03:00",  
  "threeDSecure": {  
    "url":  
    "https://localhost:55888/auth?token=Eh%2bs8NvL719YKUJ%2bbmq7xf0TWhJAm2vP1c3FDf9a9OaY  
OaYB9N%2bOSWqXIX2y9ARjr%2bWqtBzuQSkdZXOnKhecSAgjJW3S3yrc7S5leLe3JeoOsJXu8VtkdHo9v  
%2bv28OOO%2bjp2LCuL45IZzHji7fJ6thg6Cz9Z4U572NUjXtxMSVV%2b1%2fQqBRqO3qLJAduU%2fYc  
nOzl10eB3%2fgoJelC964%2bAmLSTQFs6ERXOR3fGTIJT14zGr9rhK7vynXj0NIcvXfTYJLreCXw%2bclMCg  
5gzfdfHz8tvuNjcxKZJ1VWSc%2fiH4A%3d"  
  },  
  "returnCode": "220",  
  "returnMessage": " Transaction request with authentication received. Redirect URL sent "  
}
```

Postback

Notification via application/x-www-form-urlencoded with the following transaction data:

NAME	SIZE	TYPE	DESCRIPTION
reference	Up to 16	Alphanumeric	Order code generated by the establishment
tid	Up to 20	Alphanumeric	Transaction unique identifier number
nsu	Up to 12	Alphanumeric	The sequential number returned by Rede
authorizationCode	6	Alphanumeric	Authorization number of the transaction returned by the card issuer.
date		Date	Transaction date on the format: yyyyMMdd

NAME	SIZE	TYPE	DESCRIPTION
time		Time	Transaction time on the format: HH:mm:ss
returncode	Up to 3	Alphanumeric	Transaction return code
returnMessage	Up to 256	Alphanumeric	Transaction return message
threeDSecure.returnCode	Up to 3	Alphanumeric	3DS return code
threeDSecure.returnMessage	Up to 256	Alphanumeric	3DS return message

Client MPI

The costumer MPI is used when the establishment already has a hired MPI. Therefore, to signalize the use of an already hired MPI, follow the instructions of the embedded *parameters*, contained in the table below.

In order to authenticate a transaction by the issuer with 3DS and afterwards to authorize it via e.Red, through the Client MPI, it is required for MPI service to be certified along with the card brands and Rede. Currently, the certified MPI services are: Lyra, Cardinal, and Datacash.

Transactions that use the 3DS service with client MPI, can be credit or debit type. Therefore, the type chosen must be informed in the BODY of the authorization transaction request.

POST: <https://api.userede.com.br/erede/v1/transactions>

Request:

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
threeDSecure		threeDSecure		
threeDSecure/embedded		Boolean	No	<p>Informes whether the used MPI service will be from Rede or third parties'.</p> <p>true: uses the MPI service from Rede.</p> <p>false: uses a third party MPI.</p> <p>Not sending this field will be considered "Rede MPI" use.</p>

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
threeDSecure/eci	2	Alphanumeric	Yes	Code returned to the MPI by the card brands that indicate the cardholder's authentication result along with the Issuer. Must be sent only for the 3DS authentication service use. Debit transactions must be obligatorily authenticated.
threeDSecure/cavv	Up to 32	Alphanumeric	Yes	Cryptogram code used on the transaction authentication and sent by the establishment MPI (may contain special characters). Must be sent only for the 3DS authentication service use.
threeDSecure/xid	28	Alphanumeric	*Yes	Authentication transaction ID sent by the MPI to the establishment (may contain special characters). Must be sent only for the 3DS authentication service use. The field used only for the card brand *Visa .

Response:

NAME	SIZE	TYPE	DESCRIPTION
returnCode	Up to 3	Alphanumeric	Transaction return code
returnMessage	Up to 256	Alphanumeric	Transaction return message
reference	Up to 16	Alphanumeric	Order code generated by the establishment
tid	Up to 20	Alphanumeric	Transaction unique identifier number
nsu	Up to 12	Alphanumeric	The sequential number returned by Rede
authorizationCode	6	Alphanumeric	Authorization number of the transaction returned by the card issuer.

NAME	SIZE	TYPE	DESCRIPTION
dateTime		Datetime	Transaction date on the format YYYY-MM-DDThh:mm:ss.sTZD
amount	Up to 10	Numeric	The total amount of the transaction without thousands separator and decimal separator
cardBin	6	Alphanumeric	First 6 digits of the card
last4	4	Alphanumeric	Last 4 digits of the card

Sample JSON:

Request:

```
{
  "reference": "pedido123",
  "amount": 2099,
  "cardholderName": "John Snow",
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2020,
  "securityCode": "235",
  "threeDSecure": {
    "embedded": false,
    "cavv": "BwABByIVaQAAAFwllVpAAAAAAA=",
    "eci": "05",
    "xid": "Nmp3VFdWMIewZ05pWGN3SGo4TDA="
  }
}
```

Response:

```
{
  "reference": "pedido123",
  "tid": "10073999764559129384",
  "nsu": "794728925",
  "authorizationCode": "999225",
  "dateTime": "2017-02-12T10:54:00.000-03:00",
  "amount": 2099,
  "cardBin": "544828",
  "last4": "0007",
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
```

```

{
  "method": "GET",
  "rel": "transaction",
  "href": "https://api.userede.com.br/erede/v1/transactions/10073999764559129384"
},
{
  "rel": "refunds",
  "href": "https://api.userede.com.br/erede/v1/transactions/10073999764559129384/refunds"
}
]
}

```

Zero dollar

The Zero Dollar transaction allows a previous validation to know if the card used is valid. This kind of transaction does not generate any kind of charge for the cardholder, avoiding undue debits in his balance.

The service is available for the card brands Visa and MasterCard on credit.

After the confirmation of service activation, the Zero Dollar transactions have to be sent as authorization with automatic capture (**capture = true**), **informing the value 0 on the parameter amount**.

IMPORTANT: This kind of transaction cannot be canceled.

POST: <https://api.userede.com.br/erede/v1/transactions>

Request:

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
capture		Boolean	No	Defines if the transaction will have automatic or posterior capture. Not sending this field will be considered automatic capture (true). For the Zero Dollar transaction, the parameter value must be set as true .
kind		credit / debit	No	Kind of transaction to be made. Not sending this field will be considered as credit. For Zero Dollar transaction, the parameter value must be

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
				set as credit .
reference	Up to 16	Alphanumeric	Yes	Order code informed by the establishment.
amount	Up to 10	Numeric	Yes	For the Zero Dollar transaction, send the value 0.
cardHolderName	Up to 30	Alphanumeric	No	Cardholder name printed on the card.
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 Ex.: 2021 or 21
securityCode	Up to 4	Alphanumeric	No	Card security code (usually located on the card back)

Response:

NAME	SIZE	TYPE	DESCRIPTION
reference	Up to 16	Alphanumeric	Order code generated by the establishment.
Tid	Up to 20	Alphanumeric	Transaction unique identifier number
Nsu	Up to 12	Alphanumeric	The sequential number returned by Rede

NAME	SIZE	TYPE	DESCRIPTION
authorizationCode	6	Alphanumeric	Authorization number of the transaction returned by the card issuer.
dateTime		Datetime	Transaction date on the format YYYY-MM-DDThh:mm:ss.sTZD
amount	Up to 10	Numeric	The total amount of the transaction without thousands separator and decimal separator
cardBin	6	Alphanumeric	First 6 digits of the card
last4	4	Alphanumeric	Last 4 digits of the card
returnCode	Up to 3	Alphanumeric	Transaction return code
returnMessage	Up to 256	Alphanumeric	Transaction return message

Sample JSON:

Request:

```
{
  "reference": "TZD001",
  "amount": 0,
  "cardholderName": "John Snow",
  "cardNumber": "5448280000000007",
  "expirationMonth": "12",
  "expirationYear": "2020",
  "securityCode": "235"
}
```

Response:

```
{
  "reference": "TZD001",
  "tid": "23457157041",
  "nsu": "22043",
  "authorizationCode": "112580",
  "dateTime": "2017-02-12T10:54:00.000-03:00",
  "amount": "0",
}
```

```
"cardBin": "544828",
"last4": "0007",
"returnCode": "174",
"returnMessage": "Zero dollar transaction success.",
"links": [
  {
    "rel": "self",
    "href": "http://api-hom.userede.com.br/eredev1/transactions/23457157041"
  },
  {
    "rel": "refund",
    "href": "http://api-hom.userede.com.br/eredev1/transactions/23457157041/refunds"
  }
]
}
```

Cancellation

The cancellation may be requested for all transactions, according to the instructions below:

- Authorization

The transaction cancellation procedure (without automatic capture) is allowed only for the total transaction cancellation and will have to be requested within the period stipulated for each branch. After that term, the transaction is automatically canceled.

- Capture and authorization with automatic capture

The procedure of the capture cancellation and the authorization with automatic capture may be performed partially or totally, through available channels.

On the total amount cancellation, the transaction will have status "Canceled", while on the partial cancellation, the status will be kept as "Approved", until the transaction total amount is fully canceled.

The cancellation requests may be performed in up to 7 days for the debit transactions and may vary for credit transactions, according to the branch of activity of each establishment.

For cancellations requested on the same day of the authorization transaction or transaction with automatic capture, the procedure will occur immediately, otherwise, the procedure will be performed in D+1.

For cancellations requested on the same day, only the full amount of the transaction is allowed.

POST: [https://api.userede.com.br/eredev1/transactions/\[tid\]/refunds](https://api.userede.com.br/eredev1/transactions/[tid]/refunds)

Request:

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
amount	Up to 10	Numeric	Yes	Cancellation amount without the thousands and decimal place separator. Ex.: R\$ 10,00 = 1000 R\$ 0,50 = 50

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
urls		urls	No	
Urls/kind		Alphanumeric	No	The field that identifies what's the url type: callback
urls/url	Up to 500	Alphanumeric	No	Url that will get the callback with the cancellation status after processed by Rede. It is also possible to register the url on the userede portal.

Response:

NAME	SIZE	TYPE	DESCRIPTION
returnCode	Up to 3	Alphanumeric	Transaction return code (check out the table "cancellation returns")
returnMessage	Up to 256	Alphanumeric	Transaction return message (check out the table "cancellation returns")
refundId	36	Alphanumeric	The return code of the cancellation request generated by Rede. In case the transaction is canceled by another channel that is not the API, this field will return empty.
tid	Up to 20	Alphanumeric	Transaction unique identifier number
nsu	Up to 12	Alphanumeric	The sequential number returned by Rede
refundDateTime		Datetime	Cancellation date on the format YYYY-MM-DDThh:mm:ss.sTZD
cancelId	Up to 15	Alphanumeric	Identifier code of the cancellation request transaction

JSON sample:

Request:

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

Response:

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

Notification URL

The notifications URL (callback) allows that the data of a transaction are returned via POST after the processing of the cancellations performed in D+1. The URL may be informed on the API itself or by accessing Rede portal on: e.Redre > Configurações > URL de notificações . We emphasize that in case the URL is informed on both channels, the priority of the notifications sending will always be in the one that was informed on the API.

IMPORTANT: To receive the notifications, it is required to have installed a public certificate compatible with TLS 1.2.

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

NAME	SIZE	TYPE	DESCRIPTION
type		Alphanumeric	Type of event used for the transaction: refund
tid	Up to 20	Alphanumeric	Transaction unique identifier number
nsu	Up to 12	Alphanumeric	The sequential number returned by Rede
date		Datetime	Cancellation date on the format YYYY-MM-DDThh:mm:ss.sTZD
amount	Up to 10	Alphanumeric	Cancellation amount
status	Up to 10	Alphanumeric	<ul style="list-style-type: none">• Done (Cancellation effected)• Denied (Cancellation denied)• Processing (Cancellation Processing)
cancellationNotice	Up to 15	Alphanumeric	Identifier code of the cancellation request transaction (cancellationId)
refundId	36	Alphanumeric	A return code of the cancellation request, generated by Rede. In case the transaction is canceled by another channel that is not the API, this field will return empty

Sample JSON:

Response:

```
{
  "type": "refund",
  "tid": "23457286608",
  "nsu": "4441",
  "date": "2017-05-16T19:50:20-03:00",
  "amount": 2000,
  "status": "Done",
  "cancellationNotice": "917135170239376",
  "refundId": "c21d95b5-5653-4a60-812a-249d2e0d9cca"
}
```

Transaction query

The transaction query may be performed in two ways. The first is informing the tid generated on the authorization transaction. As for the second, you have to inform the order number (reference) informed by the establishment.

Query by tid

GET: [https://api.userede.com.br/erede/v1/transactions/\[tid\]](https://api.userede.com.br/erede/v1/transactions/[tid])

Request:

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
tid	Up to 20	Alphanumeric	Yes	Transaction unique identifier number

Query by order code (reference)

GET: [https://api.userede.com.br/erede/v1/transactions?reference=\[código reference\]](https://api.userede.com.br/erede/v1/transactions?reference=[código reference])

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
reference	Up to 16	Alphanumeric	Yes	Order number generated by the establishment

Response:

NAME	SIZE	TYPE	DESCRIPTION
requestDateTime		Datetime	Requisition date on the format: YYYY-MM-DDThh:mm:ss.sTZD
authorization		authorization	
authorization/dateTime		Datetime	Authorization transaction date on the format: YYYY-MM-DDThh:mm:ss.sTZD
authorization/returnCode	Up to 3	Alphanumeric	Transaction return code

NAME	SIZE	TYPE	DESCRIPTION
authorization/returnMessage	Up to 256	Alphanumeric	Transaction return message
authorization/affiliation	Up to 9	Numeric	Establishment filiation 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	Up to 20	Alphanumeric	Transaction unique identifier number
authorization/nsu	Up to 12	Alphanumeric	The sequential number returned by Rede
authorization/authorizationCode	6	Alphanumeric	Authorization Number of the transaction returned by the card issuer.
authorization/kind	Up to 10	Alphanumeric	Payment method used on the original transaction (Credit or Debit)
authorization/amount	Up to 10	Numeric	Total purchase amount without the thousands separator. Example: 1000 = R\$10,00 R\$ 0,50 = 50
authorization/installments	Up to 2	Numeric	Installment number
authorization/cardHolderName	Up to 30	Alphanumeric	Cardholder name printed on the card.

NAME	SIZE	TYPE	DESCRIPTION
authorization/cardBin	6	Alphanumeric	First 6 digits of the card
authorization/last4	4	Alphanumeric	Last 4 digits of the card
authorization/softDescriptor	Up to 13	Alphanumeric	The message that will be displayed next to the establishment name on the cardholder's invoice.
authorization/origin	Up to 2	Numeric	Identifies the transaction origin: <ul style="list-style-type: none"> e.Redes - 1 VISA Checkout - 4 MasterPass - 6
authorization/subscription		Boolean	<p>Informs the issuer whether the transaction is coming from a recurrence. If the transaction is a recurrence, send true. Otherwise, send false.</p> <p>Not sending this field will be considered the value false.</p> <p>Rede does not manage the recurrence schedule, it only allows the retailers to indicate whether the originated transaction is from a recurrent plan.</p>
authorization/distributorAffiliation	Up to 9	Numeric	Distributor filiation number (PV)
capture		capture	
capture/dateTime		Datetime	Data da transação de captura no formato YYYY-MM-DDThh:mm:ss.sTZD
capture/nsu	Up to 12	Alphanumeric	The sequential number returned by Rede
capture/amount	Up to 10	Numeric	Capture amount
threeDSecure		threeDSecure	
threeDSecure/embedded		Boolean	Informs whether the used MPI service will be from Rede or third parties'

NAME	SIZE	TYPE	DESCRIPTION
threeDSecure/eci	2	Alphanumeric	Code returned to the MPI by the card brands that indicate the cardholder's authentication result along with the Issuer. Must be sent only for the 3DS authentication service use. Debit transactions must be obligatorily authenticated.
threeDSecure/cavv	Up to 32	Alphanumeric	Cryptogram code used on the transaction authentication and sent by the establishment MPI (may contain special characters). It must be sent only for the 3DS authentication service use.
threeDSecure/xid	28	Alphanumeric	Authentication transaction ID sent by the MPI to the establishment (may contain special characters). Must be sent only for the 3DS authentication service use. The field used only for the card brand *Visa .
threeDSecure/returnCode	3	Alphanumeric	Return code of the transaction with 3DS
threeDSecure/returnMessage	Up to 256	Alphanumeric	Return message of the transaction with 3DS
refunds		refunds	
refunds/dateTime		Datetime	Cancellation date on the format YYYY-MM-DDThh:mm:ss.sTZD
refunds/refundId	36	Alphanumeric	Return code of the cancellation request, generated by Rede.
refunds/status	Up to 10	Alphanumeric	Cancellation request status: <ul style="list-style-type: none"> • Done (Cancellation effected) • Denied (Cancellation denied) • Processing (Cancellation Processing)
refunds/amount	Up to 10	Numeric	Cancellation amount

JSON sample:

Request:

GET: <https://api.userede.com.br/erede/v1/transactions/8345000363484052380>

Or

GET: <https://api.userede.com.br/erede/v1/transactions?reference=pedido123>

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
  },
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "POST",
      "rel": "refund",
      "href": "https://api.userede.com.br/erede/v1/transactions/8345000363484052380/refunds"
    },
    {
      "method": "PUT",
      "rel": "capture",
      "href": "https://api.userede.com.br/erede/v1/transactions/8345000363484052380"
    }
  ]
}
```

Cancellation query

Used to consult a cancelled transaction. To do so, a request with either the tid or the refundId must be informed. To get a more detailed query, use the refundId.

The cancelled transactions consulting is available through the API, from Rede's portal or electronic statement file, one day after the cancellation requisition has been performed.

Cancellation query by tid

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

GET: [https://api.userede.com.br/erede/v1/transactions/\[tid\]/refunds](https://api.userede.com.br/erede/v1/transactions/[tid]/refunds)

Request:

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
tid	Up to 20	Alphanumeric	Yes	Transaction unique identifier number

Response:

NAME	SIZE	TYPE	DESCRIPTION
refundId	36	Alphanumeric	The return code of the cancellation request, generated by Rede. In case the transaction is canceled by a channel another from the API, this field will return empty
refundDateTime		Datetime	Cancellation date on the format YYYY-MM-DDThh:mm:ss.sTZD
cancelId	Up to 15	Alphanumeric	Identifier code of the cancellation request transaction returned only in D+1 requests.
status	Up to 10	Alphanumeric	Cancellation request status: <ul style="list-style-type: none">• Done (Cancellation effected)• Denied (Cancellation denied)• Processing (Cancellation Processing)
amount	Up to 10	Numeric	Cancellation amount without the thousands and decimal place separator.

JSON sample:

Request:

GET: <https://api.userede.com.br/erede/v1/transactions/23457040715/refunds>

Response:

```
{
  "refunds": [
```

```

{
  "refundDateTime": "2017-02-23T00:00:00-03:00",
  "cancellationId": "917054151307902",
  "status": "Done",
  "amount": 1
},
{
  "refundId": "a9cecd7-e040-4450-bb05-aeae0e81c1d9",
  "refundDateTime": "2017-05-18T00:00:00-03:00",
  "cancellationId": "917138153206643",
  "status": "Denied",
  "amount": 100
},
{
  "refundId": "fbfdbdd1-7bdb-4390-89f6-04ec262a93f8",
  "refundDateTime": "2017-05-25T00:00:00-03:00",
  "cancellationId": "917145112800523",
  "status": "Processing",
  "amount": 100
}
],
"links": [
{
  "method": "GET",
  "rel": "transaction",
  "href": "https://api-hom.userede.com.br/eredev1/transactions/23457040715"
}
]
}

```

Cancellation query by refundId

The cancellation query by refundId lists a specific cancellation request.

GET: [https://api.userede.com.br/eredev1/transactions/\[tid\]/refunds/\[refundId\]](https://api.userede.com.br/eredev1/transactions/[tid]/refunds/[refundId])

Request:

NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
refundId	36	Alphanumeric	Yes	The return code of the cancellation generated by Rede. If the transaction is canceled by a channel another from the API, this field will return empty, no longer being possible to query by refundId.

Response:

NAME	SIZE	TYPE	DESCRIPTION
------	------	------	-------------

NAME	SIZE	TYPE	DESCRIPTION
refundId	36	Alphanumeric	The return code of the cancellation request, generated by Rede. If the transaction is canceled by a channel another from the API, this field will return empty, no longer being possible to query by refundId.
tid	Up to 20	Alphanumeric	Transaction unique identifier number
refundDateTime		Datetime	Cancellation date on the format YYYY-MM-DDThh:mm:ss.sTZD
cancelId	Up to 15	Alphanumeric	Identifier code of the cancellation request transaction returned only in D+1 requests.
amount	Up to 10	Numeric	Cancellation amount without the thousands and decimal place separator.
statusHistory		statusHistory	
statusHistory/status	Up to 10	Alphanumeric	Cancellation request status: <ul style="list-style-type: none"> • Done (Cancellation effected) • Denied (Cancellation denied) • Processing (Cancellation Processing)
statusHistory/dateTime		Datetime	Cancellation request date on the format YYYY-MM-DDThh:mm:ss.sTZD
returnCode	Up to 3	Alphanumeric	Transaction return code
returnMessage	Up to 256	Alphanumeric	Transaction return message

JSON sample:

Request:

GET: <https://api.userede.com.br/erede/v1/transactions/105654/refunds/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",
  "status": "Processing",
  "amount": 2000,
  "statusHistory": [
    {
      "status": "Processing",
      "dateTime": "2017-01-28T08:33:00.000-03:00"
    }
  ],
  "returnCode": "00",
  "returnMessage": "Success.",
  "links": [
    {
      "method": "GET",
      "rel": "transaction",
      "href": "https://api.userede.com.br/erede/v1/transactions/9274256037511432483"
    }
  ]
}
```

SoftDescriptor

The identification on the invoice (Soft Descriptor or DBA) is a parameter that helps the cardholder on identifying the transaction generated on the card invoice.

The parameter is composed by 22 characters. The first 8 characters are meant to identify the establishment name, which is displayed in a static way on the cardholder's invoice. After 8 characters, Rede inserts a hyphen and provides more 13 characters to be sent dynamically through a transaction.

To enable this functionality, access Rede's portal on the menu e.Red > Identificação na fatura, or contact the Rede call center. If a name is not registered, the service will not be enabled.

After the service enablement via Rede's portal, the functionality will be made available within a term of up to 24 hours.

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

NAME	SIZE	TYPE	DESCRIPTION
softDescriptor	Up to 13	Alphanumeric	The message that will be displayed next to the establishment name on the cardholder's invoice.

Digital wallets

A digital wallet is an electronic solution that allows credit card's information and identity storage in a way that they can be used with safety and privacy during financial operations.

The digital wallets from Visa (Visa Checkout) and MasterCard (MasterPass) have an easy integration with our API. To hire the service, each brand must be contacted.

To indicate the use of these wallets, the following values must be informed:

NAME	SIZE	TYPE	DESCRIPTION
Origin	Up to 2	Numeric	For Visa Checkout, send value 4 For MasterPass, send value 6
storageCard	1	Alphanumeric	Inform whether the card will be stored by the establishment. 0 - transaction with a non-stored credential 1 - transaction with a credential stored for the first time 2 - transaction with an already stored credential The not sending of this field will be considered 0 (transaction with a non-stored credential)

Sample JSON:

Request:

```
{
  "reference": "pedido578",
  "amount": 2099,
  "origin": 4,
  "storageCard": "1",
  "cardholderName": "Joao das Neves",
  "cardNumber": "5448280000000007",
  "expirationMonth": 12,
  "expirationYear": 2020,
  "securityCode": "235"
}
```

Response:

```
{
  "reference": "pedido578",
  "tid": "8345000363484052380",
  "nsu": "663206341",
  "authorizationCode": "186376",
  "dateTime": "2017-02-28T08:54:00.000-03:00",
  "amount": 2099,
}
```



```

"cardBin": "544828",
"last4": "0007",
"returnCode": "00",
"returnMessage": "Success.",
"links": [
  {
    "method": "GET",
    "rel": "transaction",
    "href": "https:// api.userede.com.br/erede/v1/transactions/8345000363484052380"
  },
  {
    "method": "POST",
    "rel": "refund",
    "href": "https:// api.userede.com.br/erede/v1/transactions/8345000363484052380/refunds"
  }
]
}

```

Electronic statement

The Electronic Statement is a tool that provides the establishment the demonstration of transactions performed with credit cards, debit, and financial information, according to the conditions established on the commercial contracts.

It is composed of four files:

- EEVD - Debit Sales Electronic Statement
- EEVC - Credit Sales Electronic Statement
- EEFI - Financial Electronic Statement
- EESA – Values yet to receive/pay Statement

[Click here](#) to get more information and the manuals.

Return codes

HTTP status code

The e.Rede uses HTTP return codes to indicate the success or failure of a request from the API. The codes started with 2xx indicate success, while the codes indicated with 4xx indicate an error due to some incorrect information provided in the requisition and the codes started with 5xx indicate an error on the servers.

Success codes

RETURN	DESCRIPTION	METHOD
200	Indicates that the process was performed correctly and the return will be as expected.	GET
201	Indicates that the feature was created successfully, there must be a header location indicating the url of the new feature.	POST

RETURN	DESCRIPTION	METHOD
202	Indicates that the processing will be asynchronous, therefore, besides the header location, it'll have to return the content with a status attribute.	POST E PUT
204	Indicates that the feature was changed or deleted successfully	PUT

Error codes

RETORNO	DESCRIÇÃO
400	Badly formatted requisition.
401	Requisition requires authentication
403	Requisition denied
404	Feature not found
405	Method not allowed
408	Timeout for the requisition.
413	Requisition exceeds the maximum allowed size.
415	Invalid media type (check the requisition's content-type header)
422	Business exception
429	Requisition exceeds the maximum of calls allowed to the API

Exception launched by server(s) error

RETURN	DESCRIPTION
500	Server error

Issuer return

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

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.

returnCode	returnMessage
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.
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
174	Zero dollar transaction success.

Integration return

The integration returns are displayed whenever there is something wrong with your requisition, 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
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

returnCode	returnMessage
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
29	code: Invalid parameter format

returnCode	returnMessage
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

returnCode	returnMessage
45	number: Required parameter missing
46	installments: Not correspond to authorization transaction
47	origin: Invalid parameter format
49	The value of the transaction exceeds the authorized
50	installments: Invalid parameter format
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

returnCode	returnMessage
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
75	departureTax: Parameter should not be sent for this type of transaction
76	kind: Invalid parameter format
78	The transaction does not exist

returnCode	returnMessage
79	Expired card. The transaction cannot be resubmitted. Contact issuer.
80	Unauthorized. Contact issuer. (Insufficient funds)
82	An unauthorized transaction for the debit card.
83	Unauthorized. Contact issuer.
84	Unauthorized. The 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
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
172	Transaction already captured
173	Authorization expired
176	urls: Required parameter missing (url)

3DS return

The authenticated transactions have returns and specific 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
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

returnCode	returnMessage
259	urls: Required parameter missing
260	urls: Required parameter missing (kind/threeDSecureFailure)
261	urls: Required parameter missing (kind/threeDSecureSuccess)

Cancellation return

The canceled transactions have returns and specific messages.

returnCode	returnMessage
351	Forbidden
353	Transaction not found
354	The 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
362	RefundId not found
363	Callback Url characters exceeded 500

returnCode	returnMessage
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

Anti-fraud Rede

What is the Anti-fraud Rede

A specialized tool capable of evaluating potential fraud on the purchases of your e-commerce and inform in real time the risk level of each transaction.

To perform the fraud risk analysis, our anti-fraud system needs additional information to operate more accurately. This information will be sent as new fields (tags), in addition to the present fields in a basic requisition. These fields refer mainly to the information of the products on the shopping cart and, also, information about the buyer.

The information groups were organized in categories to make the understanding of each new field within each category easier. It follows the explanation about the groupings by category below:

- **Transaction:** Information of the financial transaction and payment data.
- **Requisition to enable anti-fraud:** Field responsible to discriminate whether or not the transaction will have the return of fraud analysis to the retailer client.
- **Buyer information:** Fields responsible for the cadastral information regarding the buyer.
- **Deliver information:** Data about the address and kind of product delivery to the buyer.
- **Billing information:** Data about the address and kind of product delivery to the buyer, which may or may not be the same delivery address.
- **Information about purchased items:** Information about the items purchased by the final consumer.
- **Device:** Information captured about the connection and device used to finish the purchase. Most of the information that comes from this block depends on the Device Fingerprint.

POST: <https://api.userede.com.br/erede/v1/transactions>

Request:

CATEGORY	NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
Transaction	reference	Up to 16	Alphanumeric	Yes	Order code generated by the establishment
	amount	Up to 10	Numeric	Yes	Transaction total amount without separator of thousands and decimals.
	cardholderName	Up to 30	Alphanumeric	No	Cardholder name printed on the card
	installments	Up to 2	Numeric	No	The number of installments in which the transaction will be authorized. From 2 to 12. The not sending of this field will be considered a payment in one lump sum (one installment)
	cardNumber	Up to 19	Alphanumeric	Yes	Card number
	expirationMonth	Up to 2	Numeric	Yes	Card expiration month
	expirationYear	2 or 4	Numeric	Yes	Card expiration year
	securityCode	Up to 4	Alphanumeric	Yes	Card security code

CATEGORY	NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
Requisition to enable the Anti-fraud	antifraudRequired		Boolean	Yes	<p>The field for anti-fraud analysis identification</p> <ul style="list-style-type: none"> To receive analysis, use True To avoid receiving analysis use False <p>The not sending of this field will be considered False</p>
Consumer information	name	Up to 60	Alphanumeric	Yes	Buyer's full name
	email	Up to 50	Alphanumeric	Yes	Buyer's e-mail
	phone			Yes	Buyer's phone data
	type	1	Alphanumeric	Yes	Phone type Parameters:1- Cellphone 2- Residential 3 - Commercial 4- Other
	ddd	3	Alphanumeric	Yes	Phone area code
	number	Up to 9	Numeric	Yes	Phone number
	gender	1		No	Buyer's gender. Parameters: M – Male F - Female
	cpf	11	Numeric	Yes	Buyer's CPF number
	documents			No	Other documents that may be sent besides CPF
	type	Up to 30	Alphanumeric	No	Other document type,

CATEGORY	NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
	number	Up to 25	Alphanumeric	No	The number of other documents.
Shipping information	addresseeName	Up to 100	Alphanumeric	No	Shipping addressee name
	type	1	Numeric	No	Shipping address type. Parameters:1 - Apartment; 2 - House; 3 - Commercial; 4 - Other
	address	Up to 100	Alphanumeric	Yes if “product”	Shipping address
	number	Up to 6	Alphanumeric	Yes	Shipping address number
	complement	Up to 50	Alphanumeric	No	Shipping address complement
	zipCode	Up to 8	Alphanumeric	Yes	Shipping address zip code
	neighborhood	Up to 60	Alphanumeric	Yes	Shipping address neighborhood.
	city	Up to 60	Alphanumeric	Yes	Shipping address city.
	state	Up to 20	Alphanumeric	Yes	Shipping address state.
Billing information	addresseeName	Up to 100	Alphanumeric	No	Name of person responsible for the billing address.
	type	1	Numeric	No	Shipping address type. Parameters:1 - Apartment; 2 - House; 3 - Commercial; 4 - Other
	address	Up to 100	Alphanumeric	Yes if “product”	Billing address

CATEGORY	NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
	number	Up to 6	Alphanumeric	Yes	Billing address number
	complement	Up to 50	Alphanumeric	No	Shipping address complement
	zipCode	Up to 8	Alphanumeric	Yes	Shipping address zip code.
	neighborhood	Up to 60	Alphanumeric	Yes	Shipping address neighborhood.
	city	Up to 60	Alphanumeric	Yes	Shipping address city.
	state	Up to 20	Alphanumeric	Yes	Shipping address state.
Information about acquired items (multiple items can be sent)	type	1	Numeric	Yes	Type of items. Only send the number corresponding to the item type Parameters: 1 - Products; 2 – Digital products; 3 - Services; 4 – Airlines
	shippingType	Up to 25	Alphanumeric	No	Shipping type
	id	10	Alphanumeric	No	Item identification for the retailer
	description	Up to 160	Alphanumeric	Yes	Purchased item description

CATEGORY	NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
	quantity	Up to 5	Numeric	Yes	Item quantity
	amount	Up to 10	Numeric	Yes	Item amount without thousands separator and decimal separator.
	freight	Up to 10	Numeric	No	Freight amount without thousands separator and decimal separator.
	discount	Up to 10	Numeric	No	Discount on items without thousands separator and decimal separator
Device	ip		Alphanumeric	No	Customer IP checker
	sessionId		Alphanumeric	No	Rede fingerprint Session ID PS: For more information, check out the item “3.Device Fingerprint”.

Response:

NAME	SIZE	TYPE	DESCRIPTION
score		Whole	Guide value for fraud decision.
recommendation	50	String	Recommendation on an action that the customer should take about the transaction. Possibilities: - Aprovar (Approve)- Negar (Deny)
risklevel	50	String	Indicates the risk level in a textual way. Possibilities - Baixo Risco (Low Risk) - Medio Risco (Medium Risk) - Alto Risco (High Risk)

NAME	SIZE	TYPE	DESCRIPTION
success		boolean	It will indicate if we have obtained a return from the fraud analysis.

Sample JSON:

```
{
  "reference": "pedido000001xpto",
  "amount": 1100,
  "cardholderName": "Pedro das Neves",
  "installments": 1,
  "cardNumber": "5448280000000007",
  "expirationMonth": 1,
  "expirationYear": 2019,
  "securityCode": "123",
  "antifraudRequired": true,
  "cart": {
    "consumer": {
      "name": "Guilherme",
      "email": "lorem@ipsum.com",
      "phone": {
        "type": 1,
        "ddd": "011",
        "number": "912341234"
      },
      "gender": "M",
      "cpf": "15153855406",
      "documents": [
        {
          "type": "RG",
          "number": "3123123123123"
        }
      ]
    },
    "shipping": [
      {
        "addresseeName": "Daenerys",
        "type": 2,
        "address": "Avenida Paulista",
        "number": "123",
        "complement": "Ap 2",
        "zipCode": "01122123",
        "neighbourhood": "Bela Vista",
        "city": "Sao Paulo",
        "state": "SP"
      }
    ],
    "billing": {
```

```

{
  "addresseeName": "Daenerys",
  "type": 3,
  "address": "Avenida Paulista",
  "number": "123",
  "complement": "Ap 2",
  "zipCode": "01122123",
  "neighbourhood": "Bela Vista",
  "city": "Sao Paulo",
  "state": "SP"
},
"items": [
  {
    "type": 1,
    "shippingType": "Sedex",
    "id": "123123",
    "description": "Televisão",
    "quantity": 1,
    "amount": 200000,
    "freight": 199,
    "discount": 199
  },
  {
    "type": 1,
    "id": "1333333",
    "description": "Geladeira",
    "quantity": 1,
    "amount": 200000,
    "freight": 199,
    "discount": 199
  }
],
"environment": {
  "consumer": {
    "ip": "192.168.137.212",
    "sessionId": " NomeEstabelecimento-WebSessionID "
  }
}
}

```

Response:

```

{
  "reference": "pedido123",
  "tid": "8345000363484052380",
  "nsu": "663206341",
  "authorizationCode": "186376",
  "dateTime": "2017-02-28T08:54:00.000-03:00",
  "amount": "2099",
  "cardBin": "544828",
  "last4": "0007",
  "returnCode": "00",

```

```

"returnMessage": "Success.",
"links": [
  {
    "rel": "self",
    "href": "http://localhost:9898/transactions/8345000363484052380"
  },
  {
    "rel": "capture",
    "href": "http://localhost:9898/transactions/8345000363484052380"
  },
  {
    "rel": "refund",
    "href": "http://localhost:9898/transactions/8345000363484052380/refunds"
  }
],
"antifraud": {
  "success": true,
  "score": -50,
  "riskLevel": "Alto Risco"
  "recomendation": "Negar"
}
}

```

Device Fingerprint

Device Fingerprint is a script written in JavaScript that performs collects and public information analysis of the user machine, as device identification, geolocation, system information, among other relevant information for a full fraud risk analysis. Besides these, information on the browsing behavior is verified to compose de model.

In which page should I include JavaScript?

For being a JavaScript responsible for capturing user actions on the website, we recommend that the page in which the script is allocated is one that represents interactivity and works as a path for all clients to pass to finish the purchase, such as filling the registering data or payment data (checkout).

It is required that the JavaScript is at least in one of the mentioned pages, however, we recommend that it is present in more than one page that is a path to the act of finishing the purchase, thus ensuring that the JavaScript is fully loaded and the analysis is the best as possible.

We recommend that the JavaScript tags inclusion be at the end of the html page, before </body>.

The fulfillment of these recommendation ensures that the website does not suffer from any loss on the integration performance. In any case, each retailer has autonomy to set the best local, according to the structure of each page.

JavaScript code

Below, the code that will have to be present on the pages, according to the structure recommendations:

```

<script type="application/javascript" src="https://fingerprint.userede.com.br/b.js"></script>
<script>
function simility(){
var sc = {
  "customer_id": "75ee1000-3f7c-11e7-af0a-d705e0dbc8bb",

```

```
"session_id": "minhaLoja-minhaSessao",  
"event_types": "checkout_page",  
"zone": "us",  
"request_endpoint": "https://fingerprint.userede.com.br"  
};  
var ss = new SimilityScript(sc);  
ss.execute();  
};  
</script>
```

Customer_id: It's a fixed value for sending, varying only on the Sandbox and Production environments, as on the table below:

Environment	CUSTOMER ID
Sandbox	c54b99d0-894e-11e7-adc9-77887d29e284
Production	75ee1000-3f7c-11e7-af0a-d705e0dbc8bb

Session_id: It's a unique value for each purchase and should not be repeated, for it will ensure the association and correct analysis of captured data with the transaction.

Suggestion: Retailer affiliation number with Rede (PV) + browser session + data.
Ex: 010101010XPTOQU20170101

Event_types: Name of the page in which the script is being placed.

Error codes

The error codes that can be displayed during Rede Anti-fraud integration are listed below:

ERROR CODE	ERROR MESSAGE
900	Transaction denied due to suspicion of fraud.
901	Consumer name is required for fraud analysis.
902	Consumer name: Invalid Parameter size.
903	Consumer e-mail is required for fraud analysis.

ERROR CODE	ERROR MESSAGE
904	Consumer e-mail: Invalid Parameter size.
905	Consumer e-mail: Invalid Parameter format.
906	Consumer CPF is required for fraud analysis.
907	Consumer CPF: Invalid Parameter size.
908	Consumer CPF: Invalid Parameter format.
909	Consumer phone type: Invalid Parameter size.
910	Consumer phone type: Invalid Parameter format.
911	Consumer phone type: Value not allowed.
912	Consumer phone number is required for fraud analysis.
913	Consumer phone number: Invalid Parameter size.
914	Consumer phone number: Invalid Parameter format.
915	Consumer phone DDD is required for fraud analysis.
916	Consumer phone DDD: Invalid Parameter size.
917	Consumer phone DDD: Invalid Parameter format.
918	Consumer gender: Invalid Parameter size.

ERROR CODE	ERROR MESSAGE
919	Consumer gender: Invalid Parameter format.
920	Consumer gender: Value not allowed.
921	Consumer document type: Invalid Parameter size.
922	Consumer document number: Invalid Parameter size.
923	Shipping address type: Invalid Parameter size.
924	Shipping address type: Invalid Parameter format.
925	Shipping address type: Value not allowed.
926	Shipping address street is required for fraud analysis.
927	Shipping address street: Invalid Parameter size.
928	Shipping address number is required for fraud analysis.
929	Shipping address number: Invalid Parameter size.
930	Shipping zip code number is required for fraud analysis.
931	Shipping zip code: Invalid Parameter size.
932	Shipping neighborhood is required for fraud analysis.
933	Shipping neighborhood: Invalid Parameter size.

ERROR CODE	ERROR MESSAGE
934	Shipping city is required for fraud analysis.
935	Shipping city: Invalid Parameter size.
936	Shipping state is required for fraud analysis.
937	Shipping state: Invalid Parameter size.
938	Shipping addressee name: Invalid Parameter size.
939	Shipping street complement: Invalid Parameter size.
940	Billing address type: Invalid Parameter size.
941	Billing address type: Invalid Parameter format.
942	Billing address type: Value not allowed.
943	Billing address street is required for fraud analysis.
944	Billing address street: Invalid Parameter size.
945	Billing address number is required for fraud analysis.
946	Billing address number: Invalid Parameter size.
947	Billing zip code number is required for fraud analysis.
948	Billing zip code: Invalid Parameter size.

ERROR CODE	ERROR MESSAGE
949	Billing neighborhood is required for fraud analysis.
950	Billing neighborhood: Invalid Parameter size.
951	Billing city is required for fraud analysis.
952	Billing city: Invalid Parameter size.
953	Billing state is required for fraud analysis.
954	Billing state: Invalid Parameter size.
955	Billing addressee name: Invalid Parameter size.
956	Billing street complement: Invalid Parameter size.
957	Item type: Invalid Parameter size.
958	Item type: Invalid Parameter format.
959	Item type: Value not allowed.
960	Item description is required for fraud analysis.
961	Item description: Invalid Parameter size.
962	Item quantity is required for fraud analysis.
963	Item quantity: Invalid Parameter size.

ERROR CODE	ERROR MESSAGE
964	Item quantity: Invalid Parameter format.
965	Item amount is required for fraud analysis.
966	Item amount: Invalid Parameter size.
967	Item amount: Invalid Parameter format.
968	Item id: Invalid Parameter size.
969	Item shipping type: Invalid Parameter size.
970	Item freight: Invalid Parameter size.
971	Item freight: Invalid Parameter format.
972	Item discount: Invalid Parameter size.
973	Item discount: Invalid Parameter format.
999	Item type is required to fraud analysis.

Fraud analysis for areas

The Anti-fraud rede has the capacity for fraud risk analysis also for Airline Companies. For that, it is required that fields beside the one mentioned on the previous items are included. Check out below, the required data for Airline Companies.

The following table informs the fields that have to be sent for Rede to perform the risk analysis with the anti-fraud exclusively for Airline Companies.PS: The column "MANDATORY" in this case is exclusive to Airline Companies.

POST: <https://api.userede.com.br/erede/v1/transactions>

Request:

CATEGORY	NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
Passenger and flight information (only for airlines)	lata				
	flight				
	date		Datetime	Yes	Flight date
	number	Up to 20	Alphanumeric	Yes	Flight number
	from	Up to 60	Alphanumeric	Yes	Flight origin
	to	Up to 60	Alphanumeric	Yes	Flight destination
	passenger ticket		Numeric	Yes	Passenger ticket number
	name	Up to 60	Alphanumeric	Yes	Passenger name
	email	Up to 50	Alphanumeric	Yes	Passenger e-mail
	phone			Yes	Passenger phone data

CATEGORY	NAME	SIZE	TYPE	MANDATORY	DESCRIPTION
	type	1	Numeric	No	Passenger phone type Parameters: 1- Phone 2- Residential 3 - Commercial 4- Other
	ddd	3	Numeric	Yes	Passenger phone DDD
	number	Up to 9	Numeric	Yes	Passenger phone number.

Sample JSON for area analysis:

```
{
  "reference": "pedido000001xpto",
  "amount": 1100,
  "cardholderName": "Joao das Neves",
  "installments": 1,
  "cardNumber": "5448280000000007",
  "expirationMonth": 1,
  "expirationYear": 2019,
  "securityCode": "123",
  "antifraudRequired": true,
  "cart": {
    "consumer": {
      "name": "Guilherme",
      "email": "teste123@teste123.com.br",
      "phone": {
        "type": 1,
        "ddd": "011",
        "number": "912341234"
      },
      "gender": "M",
      "cpf": "15153855406",
      "documents": [
        {
          "type": "RG",
          "number": "3123123123123"
        }
      ]
    }
  }
}
```

```
},
"shipping": [
{
  "addresseeName": "Daenerys",
  "type": 2,
  "address": "Avenida Paulista",
  "number": "123",
  "complement": "Ap 2",
  "zipCode": "01122123",
  "neighbourhood": "Bela Vista",
  "city": "Sao Paulo",
  "state": "SP"
}
],
"billing":
{
  "addresseeName": "Daenerys",
  "type": 3,
  "address": "Avenida Paulista",
  "number": "123",
  "complement": "Ap 2",
  "zipCode": "01122123",
  "neighbourhood": "Bela Vista",
  "city": "Sao Paulo",
  "state": "SP"
},
"items": [
{
  "type": 1,
  "shippingType": "Sedex",
  "id": "123123",
  "description": "Televisão",
  "quantity": 1,
  "amount": 200000,
  "freight": 199,
  "discount": 199
},
{
  "type": 1,
  "id": "1333333",
  "description": "Geladeira",
  "quantity": 1,
  "amount": 200000,
  "freight": 199,
  "discount": 199
}
],
"iata": {
"flight": [
{
  "date": "2017-02-15T10:54:45-9:00",
  "number": "123213",
  "from": "Los Angeles, LA",
  "to": "New York, NY",
```

```

"passenger": [
  {
    "ticket": "32423432432",
    "name": "Arya Stark",
    "email": "teste123@teste123.com.br",
    "phone": {
      "type": 1,
      "ddd": "011",
      "number": "912341234"
    }
  }
]
},
"environment": {
  "consumer": {
    "ip": "192.168.137.212",
    "sessionId": " NomeEstabelecimento-WebSessionID "
  }
}
}
}

```

Error codes for airline companies

The Airline Companies error codes that can be displayed during Rede Anti-fraud integration are listed below:

ERROR CODE	ERROR MESSAGE
974	IATA flight date is required to fraud analysis.
975	IATA flight date: Invalid Parameter format.
976	IATA flight number is required to fraud analysis.
977	IATA flight number: Invalid Parameter size.
978	IATA flight 'from' is required to fraud analysis.
979	IATA flight 'from': Invalid Parameter size.

ERROR CODE	ERROR MESSAGE
980	IATA flight 'to' is required to fraud analysis.
981	IATA flight 'to': Invalid Parameter size.
982	Flight passenger name is required to fraud analysis.
983	Flight passenger name: Invalid Parameter size.
984	Flight passenger e-mail is required to fraud analysis.
985	Flight passenger e-mail: Invalid Parameter size.
986	Flight passenger e-mail: Invalid Parameter format.
987	Flight passenger phone type: Invalid Parameter size.
988	Flight passenger phone type: Invalid Parameter format.
989	Flight passenger phone type: Value not allowed.
990	Flight passenger phone number is required to fraud analysis.
991	Flight passenger phone number: Invalid Parameter size.
992	Flight passenger phone number: Invalid Parameter format.
993	Flight passenger phone DDD is required to fraud analysis.
994	Flight passenger phone DDD: Invalid Parameter size.
995	Flight passenger phone DDD: Invalid Parameter format.

ERROR CODE	ERROR MESSAGE
996	Flight passenger ticket: Invalid Parameter size.
997	Environment consumer ip: Invalid Parameter format.
998	Environment consumer session ID: Invalid Parameter size.

Playground tutorial

To start using the integration at the sandbox it is required to create your test store informing the store name and the product that will be tested.

The test store will have a PV and a token that will have to be used only in our sandbox. With this information, you're ready to use our playground and start sending integration tests and simulating transactions.

Select your store that you want to use for tests.

You may also use this PV e Token to send the requests to the sandbox endpoint:

Sandbox	https://api.userede.com.br/desenvolvedores/v1/transactions
----------------	---

Data will be stored for 30 days.

Test cards

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

Brand	Type	Card	Expiration date	Security code
Mastercard	Debit	5277696455399733	jan/20	123
Mastercard	Credito	5448280000000007	jan/20	123
Mastercard (BIN 2)	Debit	2223000148400010	jan/20	123
Mastercard (BIN 2)	Crédito	2223020000000005	jan/20	123

Brand	Type	Card	Expiration date	Security code
Visa	Debit	4761120000000148	jan/20	123
Visa	Credito	4235647728025682	jan/20	123
Hipercard	Credit	6062825624254001	jan/20	123
Hiper	Credit	6370950847866501	jan/20	123
Diners	Credit	36490101441625	jan/20	123
JCB	Credit	3569990012290937	jan/20	123
JCB (19 dig)	Credit	3572000100200142753	jan/20	123
Credz	Credit	6367600001405019	jan/20	123
Elo	Credit	4389351648020055	jan/20	123
Amex	Credit	371341553758128	jan/20	1234
Cabal	Credit	6042034400069940	jan/20	123
Sorocred	Credit	6364142000000122	jan/20	123
Credsystem	Credit	6280281038975334	jan/20	123
Banescard	Credit	6031828795629272	jan/20	123

In case the transaction is sent with another card than the ones informed, the sandbox will return the following error:

Error code	Description
58	Unauthorized. Contact issuer.

Simulate errors

To simulate error codes, just send transactions with the amount corresponding to the error codes, as on the table below:

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.

Error code	Amount	Description
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.
107	107	Unauthorized. 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.

Error code	Amount	Description
116	116	Unauthorized. Please contact the Card Issuer.
117	117	Transaction not found.
118	118	Unauthorized. Card locked.
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.

Simulate transaction with a retroactive date

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

Transactions sent between R\$30,01 and R\$30,99 will return with the retroactive capture date of the current date. The cents numbers equal to the retroactive days.

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

Current date: 10/23

Transaction amount: 30,**03*** (amount 30**03***)

***03 refer to the retroactive days**

Generated transaction date: 10/20

Simulate Zero Dollar transaction

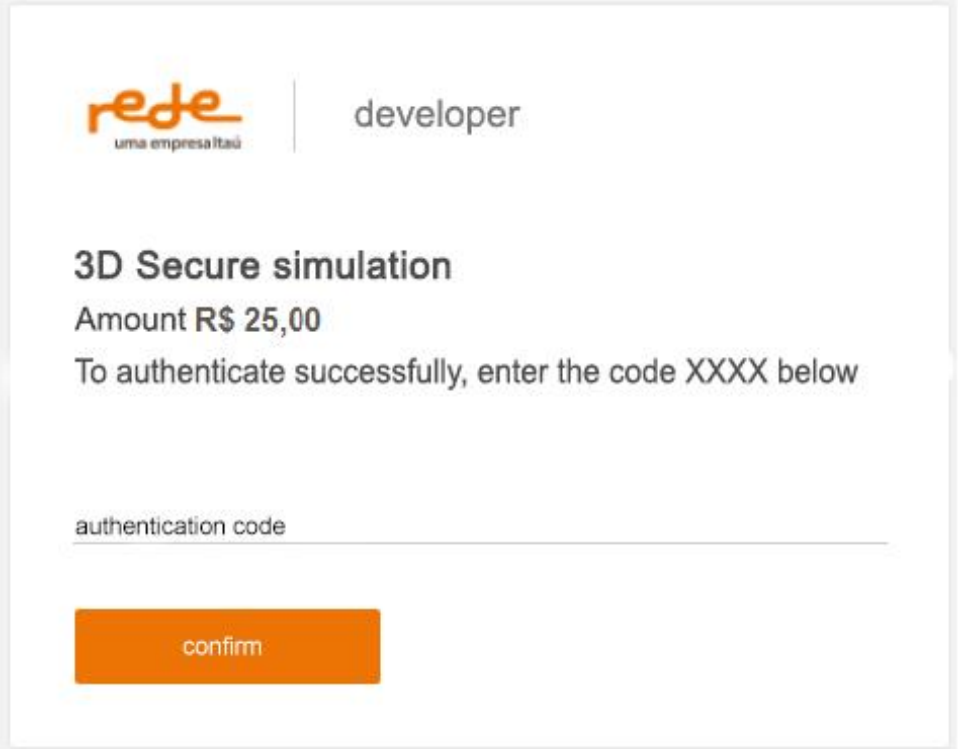
To simulate a Zero Dollar transaction, just send on the requisition the “amount: 0” and the Zero Dollar transaction will be performed.

Simulate 3DS transactions - MPI Rede

To simulate 3DS transactions with MPI Rede, send on the requisition the information referring to the threeD Secure.

The return will be “**220 – Unauthenticated cardholder**” with the URL parameter filled, copy and paste in your browser to simulate the issuer’s authentication screen, for our sandbox environment allows that it is emulated the authentication success or failure.

To simulate success, inform the code that is displayed, in case a different amount is informed, the transaction will be denied.



rede
uma empresa itaú

developer

3D Secure simulation
Amount R\$ 25,00
To authenticate successfully, enter the code XXXX below

authentication code

confirm

Attention: It’s a simulation screen. In production, the screen and information request vary according to each issuer.

View logs

All test transactions sent, are stored for 30 days in our sandbox. To view, just access your store details and you’ll be able to filter and view the details of each requisition.

Support

In case you have any question, contact us. A team will be in service for support and technical consulting.

Our technical support is only available for customers whom are already registered at Rede.

Before contacting us, please have in hand the filiation number (establishment) or the National Register of Legal Entity (CNPJ) of the company in your hand.

- Make sure you have completely read the manual and you agree with the development;
- e.Redes's technical support is not authorized to provide or receive credit or debit card numbers, valid or not, even for tests.
- e.Redes's technical support is not authorized to provide, receive or give support for source codes of the establishment or programming languages.
- e.Redes's technical support is not enabled to provide commercial information.

Contact us:

+55 11 4001 4490 (capitals and metropolitan regions)

0800 728 4490 (other location)