REST Card Service

This web service and its architecture allows any application to connect to Place to Pay to process cards, independent of the development language.

To use the web service, you will need to have a programming language that can communicate with a REST service.

The integration method contemplates the basic operations for card processing non-face-to-face, this is types of credit, interest, generation of OTP, authorization, reverse.

Each of the uses and the data required in the plot for each will be described below.

one of the cases. The response for each operation is unified in a single type of frame resulting.

By default, the language of the answers is in Spanish, if you want to change it is necessary to send the “locale” parameter in requests.

How does it work?

It is a process to carry out the transaction that involves the consumption of services.

described in this document.

1. Once the user's card number is captured, consumption is made to the service of information (information Request) by sending it in the request

2. The response of the service is analyzed to provide the options that follow

a. If the response brings types of credit "credits" must be shown to the user to allow you to select which one you want to use.

b. If the response has the "display Interest" parameter set to true, it must, once selected a type of credit, use the interest calculation service (Interest Calculation) and show the user the answer, so every time the user Change your type of credit.

c. If the response brings the "require" parameter to true, the consumption must be done to the OTP generation service (otp Generation) and obtain the code from the user

which will be 6 digits to send it later in the validation service.

OTP (Otp Validate) once there is a response, the result will be sent

signature under the OTP field of the processing request.

3. Once the credit type (if any) has been earned, the OTP (if required) is consumed

the transaction processing service (process Transaction) and parses the response

that it can be final (approved, declined or failed) or that it can be pending. In

both cases will return a value for internal Reference which is very important

store because it will allow you to make subsequent queries with the service

(Query Transaction).

a. If the transaction was approved it is important to also store the number of

authorization, will allow you to make reversals later if necessary.

b. Transactions are generally completed in a short time (less than 3

sec), but if the transaction is left in pending state, it is important to have a

croon job or similar that is consulting every 5 minutes for transactions

pending until resolved.

When is it better to use it?

This mechanism should only be used by those clients of the platform that require

capture cardholder information including sensitive data.

Integration is recommended only in cases where it is not feasible for the user to perform

the transaction by entering sensitive data in Place to Pay or where a control is required

specific to the operation.

Some examples are:

● Recurring charges from a proprietary database.

● Integration with an audio response system.

● Multiple transactions to different collectors with the same information of the

cardholder.

● Proprietary mobile interface, not based on HTML.

● Integration with systems whose entry of sensitive data is not carried out directly by the

Cardholder

What obligations do I have when I use this integration?

Note that by using this type of integration, you are supplying all the data of the buyer including the sensitive data of the card, this fact alone requires that the capture of the information is by a secure mechanism that does not endanger the data of the cardholder. If the information is captured by a Web page, the point where it is captured this data must be over secure HTTPS protocol using security certificates issued by an accredited certifying entity.

If the capture is carried out by a Call Center, the person who obtains the sensitive data must be a IVR (Interactive Voice Response) or audio response mechanism; this would imply a break in the process between the process carried out by the operator with his CRM and the IVR who Capture sensitive data. Finally, the CRM consolidates the two data sets and sends them to Place to Pay.

If the capture comes from a previously registered card database, make sure that this information is encrypted by a secure encryption mechanism such as AES or 3DES, decrypt only when building the frame to be sent to Place to Pay. Place to Pay has tokenization services that can be used.

Please note that as a general rule you should not store the card number, date expiration and CVV2. These can only have persistence during the request of the authorization, once processed should not be stored.

Like any integration process, this will require certification from support staff of Place to Pay to review issues of operation, best practices and usability.

Services

All consumption is made to the base URL, which will be provided by the operations area and the endpoint of the service to be consumed is added to said URL.

For example, if the base URL were https://placetopay.com. to consume the service, the request for information a POST is made with the example data to the URL https://placetopay.com/gateway/information.

Information Request

POST /gateway/information

This service provides information about the user's card to be processed, such as what services will be used for her and the types of credit, if applicable, what is there for her, yes there are no credit types an empty array is returned and if there are, each credit type must be iterated with the installments found in the fix.

In the example of the answer there is a type of credit "DIF PLUS PAGO TOTAL" and it has 3, 6, 9 ,12 in the installments array, the user should be given the option to select

DIF PLUS TOTAL PAYMENT (3) months

DIF PLUS TOTAL PAYMENT (6) months

DIF PLUS TOTAL PAYMENT (9) months

DIF PLUS TOTAL PAYMENT (12) months

And so on, this is an example of how they are displayed in the redirection service in a Select

Request example


```
{
  "auth": {
  "login": "6dd490faf9cb87a9862245da41170ff2"
  "tranKey": "/K9f7yJpyptkPrADFWoWYsO5yZU=",
  "nonce": "WW1aa01XWXlOVEEzT1RoalptSmpOalEzT1dNd05UTXlZMkUwWlRRME1UVT0="
"seed": "2018-01-29T17:09:38-05:00"
  },
  "instrument": {
  "card": {
  "number": "36545400000008"
  }
  },
```

```
 "payment": {
 "reference": "testing_12",
 "amount": {
 "full": 12.10,
 "currency": "USD"
 }
 }
}
```


Interest Calculation

POST/GATEWAY/INTERESTS

This service must be consumed if the card requires interest to be shown

(displayInterest in true) and as an example in this way the values are displayed in the service

Redirect.

Request example


```
{
 "auth": {
 "login": "6dd490faf9cb87a9862245da41170ff2",
 "tranKey": "/K9f7yJpyptkPrADFWoWYsO5yZU=",
 "nonce": "WW1aa01XWXlOVEEzT1RoalptSmpOalEzT1dNd05UTXlZMkUwWlRRME1UVT0=",
 "seed": "2018-01-29T17:09:38-05:00"
 },
 "instrument": {
 "card": {
 "number": "36545400000008"
 },
 credit: {
 "code": 1,
```

```
 "type": "22",
 "groupCode": "M",
 "installation": 12
 }
 },

"payment": {
"reference": "testing_12",
"amount": {
"full": 120.01,
"currency": "USD"
}
}
}
```


OTP Generation

POST/GATEWAY/otp/generate

This service is consumed if the otp is necessary for the card provided by the customer (require to true) and the user should be allowed to enter the OTP to send later in the processing service, as an example in this way it is captured in the interface of Redirect.

Request example


```
{
 "auth": {
 "login": "6dd490faf9cb87a9862245da41170ff2",
 "tranKey": "+ZmLlE0ZgHa/C9wxTfDaHHIjb4krPxj/zQzpJsmI81w=",
 "nonce": "TldFeFpqZGpPRE5rTVRNd1pUUTJPREZoTTJNeE1UYzFOekJsWVRVNFl6TT0=",
 "seed": "2018-03-06T10:43:36-05:00"
```

```
},
"instrument": {
"card": {
"number": "36545400000008"
}
},
"payment": {
"reference": "testing_12",
"amount": {
"full": 120.10,
"currency": "USD"
}
}
}
```


OTP Validation (otpValidation)

POST/gateway/otp/validate

It allows validating that the OTP provided matches the one sent by the provider, if so, it is will return a "signature" value that must be sent in the processing request as "otp" this in order to be taken into account when analyzing the transaction by the filters of Safety.

Request example


```
{
"auth": {
"login": "6dd490faf9cb87a9862245da41170ff2",
"tranKey": "/K9f7yJpyptkPrADFWoWYsO5yZU=",
"nonce": "WW1aa01XWXlOVEEzT1RoalptSmpOalEzT1dNd05UTXlZMkUwWlRRME1UVT0=",
"seed": "2018-01-29T17:09:38-05:00"
},
"instrument": {
```

```
 "card": {
 "number": "36545400000008"
 },
 "otp": "123456"
 },
 "payment": {
 "reference": "TEST_20171108_144400",
 "amount": {
 "full": 120.10,
 "currency": "USD"
 }
 }
}
```


Transaction processing (process Transaction)

POST/GATEWAY/PROCESS

This service allows the user's card to be charged, the parameters of the instrument are variables, if the type of credit or otp is not requested, it is not necessary to send those variables, payer is always required, buyer is optional but recommended.

For certification reasons, especially with Interdin it is necessary to show it once finished the process the response to the user, this is an example of how it is shown in redirection, the form does not matter as much as the data that must be present.

Request example


```
{
 "auth": {
 "login": "45974e82a3867355d3471bc4a1f18a1c",
 "tranKey": "/K9f7yJpyptkPrADFWoWYsO5yZU=",
 "nonce": "WW1aa01XWXlOVEEzT1RoalptSmpOalEzT1dNd05UTXlZMkUwWlRRME1UVT0=",
 "seed": "2018-01-29T17:09:38-05:00"
```

```
  },
  "locale": "es_EC",
  "payment": {
  "reference": "TEST_20171108_144400",
  "description": "Ipsam quia sunt dolore minus atque blanditiis corrupti.",
"amount": {
  "taxes": [
  {
  "kind": "ice",
  "amount": 4.8,
  "base": 40
  },
  {
  "kind": "valueAddedTax",
  "amount": 7.6,
  "base": 40
  }
  ],
  "details": [
  {
  "kind": "shipping",
  "amount": 2
  },
  {
  "kind": "tip",
  "amount": 2
  },
  {
  "kind": "subtotal",
```

```
"amount": 40
}
],
"currency": "USD",
"overall": 56.4
}
},
"ipAddress": "127.0.0.1",
"userAgent": "Mozilla/5.0 USER_AGENT HERE",
"additional": {
"SOME_ADDITIONAL": "http://example.com/yourcheckout",
},
"instrument": {
"card": {
"number": "36545400000008",
"expirationMonth": "12",
"expirationYear": "21",
"cvv": "123"
},
credit: {
"code": "1",
"type": "02",
"groupCode": "P",
"installment": "24"
},
"otp": "a8ecc59c2510a8ae27e1724ebf4647b5"
},
"payer": {
"document": "8467451900",
```

```
 "documentType": "CC",
 "name": "Miss Delia Schamberger Sr.",
 "surname": "Wisozk",
 "email": "tesst@gmail.com",
 "mobile": "3006108300"
 },
 "buyer": {
 "document": "8467451900",
 "documentType": "CC",
 "name": "Miss Delia Schamberger Sr.",
 "surname": "Wisozk",
 "email": "tesst@gmail.com",
 "mobile": "3006108300"
 }
}
```


Safe transaction processing (safeProcessTransaction)

POST /gateway/safe-process

Processing method that requires the OTP to be sent, can be sent as the endpoint previous (Previously validated) or you can send the value sent by the user so that it is done internal validation, should only be used with cards that in the service of obtaining information require OTP.

Its parameters are the same as the transaction processing service.

Request example


```
{
 "auth": {
 "login": "45974e82a3867355d3471bc4a1f18a1c",
 "tranKey": "/K9f7yJpyptkPrADFWoWYsO5yZU=",
"nonce": "WW1aa01XWXlOVEEzT1RoalptSmpOalEzT1dNd05UTXlZMkUwWlRRME1UVT0=",
 "seed": "2018-01-29T17:09:38-05:00"
```

```
},
"locale": "es_EC",
"payment": {
"reference": "TEST_20171108_144400",
"description": "Ipsam quia sunt dolore minus atque blanditiis corrupti.",
"amount": {
"taxes": [
{
"kind": "ice",
"amount": 4.8,
"base": 40
},
{
"kind": "valueAddedTax",
"amount": 7.6,
"base": 40
}
],
"details": [
{
"kind": "shipping",
"amount": 2
},
{
"kind": "tip",
"amount": 2
},
{
"kind": "subtotal",
```

```
"amount": 40
}
],
"currency": "USD",
"overall": 56.4
}
},
"ipAddress": "127.0.0.1",
"userAgent": "Mozilla/5.0 USER_AGENT HERE",
"additional": {
"SOME_ADDITIONAL": "http://example.com/yourcheckout",
},
"instrument": {
"card": {
"number": "36545400000008",
"expirationMonth": "12",
"expirationYear": "21",
"cvv": "123"
},
credit: {
"code": "1",
"type": "02",
"groupCode": "P",
"installment": "24"
},
"otp": "123456"
},
"payer": {
"document": "8467451900",
```

```
 "documentType": "CC",
 "name": "Miss Delia Schamberger Sr.",
 "surname": "Wisozk",
 "email": "tesst@gmail.com",
 "mobile": "3006108300"
 },
 "buyer": {
 "document": "8467451900",
 "documentType": "CC",
 "name": "Miss Delia Schamberger Sr.",
 "surname": "Wisozk",
 "email": "tesst@gmail.com",
 "mobile": "3006108300"
 }
}
```


Transaction query (queryTransaction)

POST /gateway/query

Request example


```
{
 "auth": {
 "login": "45974e82a3867355d3471bc4a1f18a1c",
 "tranKey": "/K9f7yJpyptkPrADFWoWYsO5yZU=",
 "nonce": "WW1aa01XWXlOVEEzT1RoalptSmpOalEzT1dNd05UTXlZMkUwWlRRME1UVT0=",
 "seed": "2018-01-29T17:09:38-05:00"
 },
 "internalReference": 1449216982
}
```


Reverse Transaction (reverseTransaction)

POST /gateway/transaction

Request example


```
{
  "auth": {
  "login": "45974e82a3867355d3471bc4a1f18a1c",
  "tranKey": "/K9f7yJpyptkPrADFWoWYsO5yZU=",
  "nonce": "WW1aa01XWXlOVEEzT1RoalptSmpOalEzT1dNd05UTXlZMkUwWlRRME1UVT0=",
  "seed": "2018-01-29T17:09:38-05:00"
  },
  "internalReference": 1449217014,
  "authorization": "000000",
  "action": "reverse"
}
```


Tokenization (tokenize)

POST /gateway/tokenize

This service allows you to store credit cards safely, through a request that contains the information of the same, a token will be generated that can be used in the processing service and for all purposes, in PlacetoPay it is equivalent to a credit card credit, the difference is that the token structure is sent instead of card.

Before using this service, you should consult the information section to find out if it is necessary or not to generate an OTP to the client and, if necessary, request the token from the person and send it in consumption.

Request example


```
{
  "auth": {
  "login": "6dd490faf9cb87a9862245da41170ff2",
  "tranKey": "xkDytkHMxIqiY1cCk+QjBXS0QjM=",
  "nonce": "T0RRM05UWTRNV0kyTWpJM1ptRXdZVFZoWmpnd1lqWmpNV0UzWW1VMU5EST0=",
```

```
 "seed": "2017-11-01T16:23:21-05:00"
 },
 "payer": {
 "name": "Dr Wilfredo Cole Sr.",
 "surname": "Corwin",
 "email": "jewell32@abbott.com"
 },
 "instrument": {
 "card": {
 "number": "4111111111111111",
 "expirationMonth": "02",
 "expirationYear": "20",
 "cvv": "123"
 },
 "otp": "123456"
 },
 "ipAddress": "127.0.0.1",
 "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36"
}
```


Transaction search (search)

POST /gateway/search

This service allows the search of transactions by reference and amount, it is usually used in communication loss case when creating a transaction to get the internal reference Again.

Request example


```
{
 "auth": {
 "login": "6dd490faf9cb87a9862245da41170ff2",
```

```
 "tranKey": "Sp3zZPnK\/A4kQxcwqEIN3c4erol3bLPTzJQVJwwNBlw=",
 "nonce": "NjU1MTAyMw==",
 "seed": "2018-04-18T04:04:40+00:00"
 },
 "reference": "TEST_20180516_182751",
 "amount": {
 "currency": "USD",
 "full": 32.43
 }
}
```


Data structures

authentication

Structure containing the site authentication information generated according to the WSSE UsernameToken Profile 1.1

1. Login
2. Trankey
3. Nonce
4. Seed

Status

Structure that contains the information about a request or payment, informs the current status of the same.

1. Status
2. Reason
3. Message
4. Date

payment

Structure that contains the information about the payment of the transaction required to the web service.

1. Reference
2. Description
3. Amount
4. Dispersion

Dispersion

Structure containing the information required for a scatter payout.

1. Nombre
2. Agreement

11.1. AgreementType

11.2. amount

agreementType

Indicates the type of agreement with which the dispersion will be applied

AIRLINE applies to air ticket transactions, where two transactions are made, one at a time.
name of the airline (in this the total value of the ticket is determined) and another in the name of the travel agency (The value of the travel agency fee is determined).

Tokenization (tokenize)

POST /gateway/tokenize

This service allows you to store credit cards safely, through a request
that contains the information of the same, a token will be generated that can be used in the processing service and for all purposes, in PlacetoPay it is equivalent to a credit card credit, the difference is that the token structure is sent instead of card.
Before consuming this service, you should consult the information section to find out if it is necessary or not to generate an OTP to the client and, if necessary, request the token from the person and send it in consumption.

Request example


```
{
  "auth": {
  "login": "6dd490faf9cb87a9862245da41170ff2",
  "tranKey": "xkDytkHMxIqiY1cCk+QjBXS0QjM=",
  "nonce": "T0RRM05UWTRNV0kyTWpJM1ptRXdZVFZoWmpnd1lqWmpNV0UzWW1VMU5EST0=",
  "seed": "2017-11-01T16:23:21-05:00"
  },
```

```
 "payer": {
 "name": "Dr Wilfredo Cole Sr.",
 "surname": "Corwin",
 "email": "jewell32@abbott.com"
 },
 "instrument": {
 "card": {
 "number": "4111111111111111",
 "expirationMonth": "02",
 "expirationYear": "20",
 "cvv": "123"
 },
 "otp": "123456"
 },
 "ipAddress": "127.0.0.1",
 "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36"
}
```


Transaction search (search)

POST /gateway/search

This service allows the search of transactions by reference and amount, it is usually used in communication loss case when creating a transaction to get the internal reference again.

Request example


```
{
 "auth": {
 "login": "6dd490faf9cb87a9862245da41170ff2",
 "tranKey": "Sp3zZPnK\/A4kQxcwqEIN3c4erol3bLPTzJQVJwwNBlw=",
```

```
 "nonce": "NjU1MTAyMw==",
 "seed": "2018-04-18T04:04:40+00:00"
 },
 "reference": "TEST_20180516_182751",
 "amount": {
 "currency": "USD",
 "full": 32.43
 }
}
```


Data structures

authentication

Structure containing the site authentication information generated according to the WSSE Username Token Profile 1.1

1. Login
2. Trankey
3. Nonce
4. seed

Status

Structure that contains the information about a request or payment, informs the current status of the same.

1. Status
2. Reason
3. Message
4. date

payment

Structure that contains the information about the payment of the transaction required to the web service.

1. Reference
2. Description
3. Amount
4. Dispersion

Dispersion

Structure containing the information required for a scatter payout.

1.Nombre

2. agreement

3.agreementType

4. amount

agreement Type

Indicates the type of agreement with which the dispersion will be applied

AIRLINE: applies to air ticket transactions, where two transactions are made, one at a time.

name of the airline (in this the total value of the ticket is determined) and another in the name of the travel agency (The value of the travel agency fee is determined)

airline agreements

Example of requests

```
'payer' => [
 'name' => 'Diego',
 'email' => 'diego.calle@placetopay.com',
 ],
'payment' => [
 'reference' => 'ON1434012-PN1433129',
 'description' => 'Flight 50 AV2020',
 'amount' => [
 'taxes' => [
 [
 'kind' => 'airportTax',
 'amount' => 16300,
 ],
 [
 'kind' => 'valueAddedTax',
 'amount' => 15847,
 ],
```

```
  ],
  'currency' => 'COP',
  'total' => 116112,
  ],
  'scatter' => [
//Airline transaction information
  [
  'agreement' => 19,
  'agreementType' => 'AIRLINE',
  'amount' => [
  'taxes' => [
  [
  'kind' => 'airportTax',
  'amount' => 16300,
  ],
[
  'kind' => 'valueAddedTax',
  'amount' => 14250,
  ],
  ],
'currency' => 'COP',
  'total' => 105550,
  ],
  ],
//Travel agency transaction information
  [
  'agreement' => null,
  'agreementType' => 'MERCHANT',
  'amount' => [
```

```
  'taxes' => [
  [
  'kind' => 'valueAddedTax',
  'amount' => 1597,
],
  ],
'currency' => 'COP',
  'total' => 10562,
  ],
  ],
  ],
  ],
  'instrument' => [
  'card' => [
  'number' =>'xxxxxxxxxxxxxxxx' ,
  'expiration' => 'xx/xx',
  'cvv' => 'xxx',
  ],
  ],
  'ipAddress' => '127.0.0.1',
  'userAgent' => 'Testing',
];
```

MERCHANT: applies to transactions where funds are dispersed between establishments that are enabled in Placetopay

Amount

Structure that contains the information about the payment of the transaction required to the web service.

1. Currency
2. Total
3. Taxes

4. Details

TaxDetail

Structure to store information about tax.

1. Kind
2. Amount
3. base

AmountDetail

Structure to store information about an additional value.

1.kind

2. amount

instrument

Structure that contains the information about the means of payment to be used in a transaction, this structure

It is variable according to the request that is generated, each service requires that one or the other data be used.

1.card

2. token

3.credit

4.otp

Cards

Structure containing card information.

1.number

2.expiration

3.expiration Year

4. cvv

Token

Structure that contains the information of the token that associates to the card.

1.token

2.subtoken

3. franchise

4.franshiseName

5.issuerName

6.lastDigits

7.validUntil

Credit

Structure that contains the information of the type of credit.

1.code

2.type

3.groupCode

4.installment

5.installments

InterestValues

Structure that contains the interest calculation information for a type of credit and amount.

1.original

2.installment

3.interest

4.total

AmountConversion

Structure to define the conversion factor and values.

1.from

2.to

3.factor

AmountBase

Structure that represents a quantity that defines the currency and the total.

1.currency

2.total

people

Structure that reflects the information of a person involved in a transaction.

1. documentType
2. document
3. name
4. surname
5. company
6. email
7. address
8. mobile

address

Structure containing information about a physical address.

1.street

2.city

3.state

4.postalCode

5.country

6.phone

transaction

Structure that contains the information of a transaction.

1. status
2. provider
3. internalRefernce
4. reference
5. paymentMethod
6. franchise
7. franchiseName
8. lisuerName
9. amount
10. conversion
11. authorization
12. receipt
13. type
14. refunded
15. lastDigits
16. additional

Authentication

Authentication example:

All web requests must be authenticated with Web Services Security UsernameToken

Profile 1.1.

https://www.oasis-open.org/committees/download.php/13392/wss-v1.1-spec-pr UsernameTokenProfile-01.htm

Using PasswordDigest

(PasswordDigest = Base64 (SHA-256 (nonce + seed + tranKey)))

Example

To connect with the REST API, the auth must be merged as auth parameter with the

request data.

Provided information:

login = testuser

tranKey = ABCD1234

User Generated Data

Random value for each request

nonce = 12345678

ISO 8601 date (Y-m-d\TH:i:s\Z) or (Y-m-d\TH:i:sP)

seed = 2018-01-29T17:02:49-05:00

Note that nonce and seed are generated, login and trankey are provided by Place to

Pay, with this information the data to be sent must be:

Base64(SHA-256(nonce + seed + tranKey))

tranKey = dsRL5wIymrySr9TgsYtxWZEIb5/RtW1v3n3xLqQZKj4=

Base64(Nonce)

nonce = MTIzNDU2Nzg=

The resulting trankey value is Base64(SHA-256(nonce + seed + tranKey))

The nonce inside SHA-256 is not Base64-encoded, it is only parameter-encoded.

The structures in REST are the same as described above, but are

encoded in JSON.

{

```
 "auth": {
 "login": "testuser",
 "tranKey": "dsRL5wIymrySr9TgsYtxWZEIb5/RtW1v3n3xLqQZKj4=",
 "nonce": "MTIzNDU2Nzg=",
 "seed": "2018-01-29T17:02:49-05:00"
 }
}
```


Problem solving

Response codes for failed authentication and what they mean

Additional Information

<Test card numbers

It is important to define that these cards only have this behavior in the environment of

Tests.

Franquicia

Numero

Comportamiento

Means of payment codes

List of payment method codes.

Codigo

Medio de pago

Document types

This is a list of the types of documents accepted by redirection.

Pais

Codigo

Tipo de documento

*** For Ecuador Redirection, only CI, RUC, PPN are enabled**

Service Response Codes

List of response codes for Interdin

Codigo

Detalle