

Overview

Submit Charge API allows you to submit a payment transaction to collect money from the customer. The API requires HMAC signature, please refer to *API Authentication* for more information about the signature.

API Endpoint	https://{server.com}/api/v3/charges
Method	POST
Content Type	Application-json

Payment Flow

1. The customer chooses *SPEI* as a payment method
2. The merchant submits the payment request to PayAmigo platform
3. The PayAmigo platform process a payment transaction
4. The PayAmigo platform responds to the merchant, with transaction status “CHARGE_PENDING” and instructions for the payer (as an option, iframe with instructions)
5. The PayAmigo platform notifies the merchant the final payment result after the payer completes the payment.

Refer to *Real-Time Notifications* for more information about the notification.

Request Parameters

Field	Required	Description
command	Y	Fixed as “CHARGE”
id	Y	1-100 chars, letter, number and hyphen are allowed. The unique transaction ID specified by the merchant. It fails if a 2nd charge request with the same transaction ID
customer.id	Y	1-100 chars, letter,number and hyphen are allowed. Customer id specified by the merchant. For a same customer please use the same ID, it helps PayAmigo to identify fraud

customer.firstName	Y	string of length 0 to 50. First name of customer
customer.lastName	Y	string of length 0 to 50. Last name of customer
customer.dateOfBirth	N	string of length 0 to 10. Customer date of birth in yyyy-mm-dd format
payment.paymentMethod	Y	Fixed as "SPEI" in uppercase.
payment.countryCode	Y	The ISO country code of the customer Mexico: MX
payment.documentType	Y	String. RFC or CURP. This should be the document type the Payer used to open his/her bank account. Generally for consumers it is CURP and for businesses it is RFC.
payment.document	Y	String. RFC number or CURP number. See last section in this document for details. RFC number is 10 to 13 digit number. CURP is an 18 character alphanumeric code. The merchant should not pass any special characters in the payload
payment.accountHolder	Y	Payer full name
payment.accountNumber	N	Payer's CLABE
metadata.callbackURL	N	The callback URL to receive real-time notifications. If it is not present, notifications are sent to the url configured in merchant's profile

Sample Request

```
{
  "command": "CHARGE",
  "customer": {
    "id": "testUser",
    "firstName": "Jane P",
    "lastName": "Doe"
  },
  "lineItem": {
    "currency": "MXN",
    "amount": 12,
    "description": "5 Golden Rings"
  }
}
```

```
"payment": {
  "paymentMethod": "SPEI",
  "countryCode": "MX",
  "accountHolder": "Jane P Doe",
  "accountNumber": "148010298543151565",
  "documentType": "RFC",
  "document": "PDJ130815TWA"
},
"metadata": {
  "callbackURL": "https://merchant.com/listener/for/callback"
}
}
```

Response Fields

To validate a payment is successful, you can follow these steps:

1. Check HTTP status, should be 201
2. Check the \$.status field in json response. Should be "CHARGE_PENDING"

Field	Description
id	The unique transaction ID specified by the merchant.
internalId	The unique PayAmigo transaction ID generated by the system.
status	Transaction status, possible values are: CHARGE_PENDING -- the payment is submitted ERROR -- the payment has failed DECLINED -- the payment is declined by the gateway, it has failed.
merchantAccount	The merchant account name
lineItem	Echo the lineItem information submitted by the merchant
instruction	Contains information about the payment that needs to be displayed to the customer to complete the payment process.
instruction.paymentType	Payment type: BANK_TRANSFER
instruction.accountType	SPEI account identifier: "CLABE"
instruction.accountNumber	CLABE number

instruction.beneficiaryName	Recipient name
instruction.expireTime	Expiration of the payment. Customer needs to complete the transaction before the expiration time.
instruction.reference	Reference of the transaction. Customer must include it as identifier of the transaction.
transactions	An array of transactions denoting the transaction activities
transactions[n].timestamp	The time of the activity performs, in UTC
transactions[n].	Historical details about this charge
transactions[n].gatewayProvidedId	The transaction ID on Acquirer side
transactions[n].type	Activity type, must be "CHARGE"
transactions[n].state	Activity status PENDING – the transaction is submitted APPROVED - the transaction is completed ERROR - the transaction is failed with error, please refer to 'resultCode' & 'responseText' for the error detail. DECLINED - the transaction is declined
transactions[n].amount	Activity amount
transactions[n].currency	Activity currency in ISO code format
transactions[n].resultCode	Activity result code Refer to " <i>Hard Decline & Result Codes</i> " for more detail
transactions[n].responseText	Description of the result code.
paymentType	It is "BANK_TRANSFER"
paymentMethod	Customer method of payment. SPEI
created	Time this charge was first submitted in ISO-8601 format
paymentOption.type	Fixed as "OFFSITE". Indicates that a customer needs to be redirected to offsite to complete the payment.
paymentOption.uiType	Fixed as "IFRAME". Indicate that the off site has no constraints and should render properly regardless of if it is displayed in an iframe.
paymentOption.offsiteMethod	Fixed as "GET". Determines if the paymentServiceURL should be redirected to with a HTTP GET.
paymentOption.paymentServiceU	URL to where you should redirect (HTTP GET) the

RL

customer to complete the payment.

Sample Response for SPEI

Refer to \$.status, if it is "CHARGE_PENDING", show instructions to the payer

```
{
  "id": "cs_69efa6ffb501478b90575f1b19a15e59",
  "internalId": "13267668581617664",
  "status": "CHARGE_PENDING",
  "merchantAccount": "DemoStore",
  "lineItem": {
    "amount": 12,
    "currency": "MXN",
    "description": "5 Golden Rings",
    "taxCode": "NT",
    "taxIncludedInPrice": true
  },
  "properties": {
    "PAYER_DOCUMENT": "PDJ130815TWA"
  },
  "instruction": {
    "paymentType": "BANK_TRANSFER",
    "accountType": "CLABE",
    "beneficiaryName": "DemoStore",
    "accountNumber": "148010230897662140",
    "expireTime": "2025-03-18 22:11:18 CST",
    "reference": "aFnA2879601R0Qs"
  },
  "transactions": [
    {
      "timestamp": "2025-03-18T03:11:19Z",
      "gatewayProvidedId": "148010230897662140",
      "type": "CHARGE",
      "state": "PENDING",
      "amount": 12,
      "currency": "MXN",
      "resultCode": 0,
      "responseText": "Success."
    }
  ],
  "paymentType": "BANK_TRANSFER",
  "paymentMethod": "SPEI",
  "created": "2025-03-18T03:11:18Z",
  "amountRemaining": 0,
  "paymentOption": {
    "type": "OFFSITE",
    "offsiteMethod": "GET",
    "uiType": "IFRAME",
  }
}
```

```
"paymentServiceURL":  
"https://sandbox.payamigo.com/pwthree/launch?payload=%7B%22timestamp%22%3A1742267479%2C%22merchantAccount%22%3A%22Kraken%22%2C%22mode%22%3A%22PROXY%22%2C%22poId%22%3A2879601%2C%22chargeId%22%3A%22cs_69efa6ffb501478b90575f1b19a15e59%22%2C%22customerLocale%22%3A%22en_US%22%7D&signature=7b3cc33f3b992ec41f06a7d8f5009f4a3ecd0794c65d712929b4221cfedb459c"  
}  
}
```

Sample Error Response

If you receive \$.status = "ERROR" or "DECLINED", the payment is failed. Refer to transactions[0].responseText for the detailed error reason.

```
{  
  "id": "cs_a97d0c8b6cd34a94ae066ac78c101edc",  
  "internalId": "13895881073238016",  
  "status": "ERROR",  
  "merchantAccount": "Merchant",  
  "lineItem": {  
    "amount": 12.00,  
    "currency": "BRL",  
    "description": "5 Golden Rings",  
    "taxCode": "NT",  
    "taxIncludedInPrice": true  
  },  
  "transactions": [  
    {  
      "timestamp": "2025-05-12T14:32:39Z",  
      "type": "CHARGE",  
      "state": "ERROR",  
      "amount": 12,  
      "currency": "BRL",  
      "resultCode": 1170,  
      "responseText": "There are required inputs missing or invalid. 'document'"  
    }  
  ],  
  "paymentType": "WALLET",  
  "paymentMethod": "PIX",  
  "created": "2025-05-12T14:32:39Z",  
  "amountRemaining": 0  
}
```

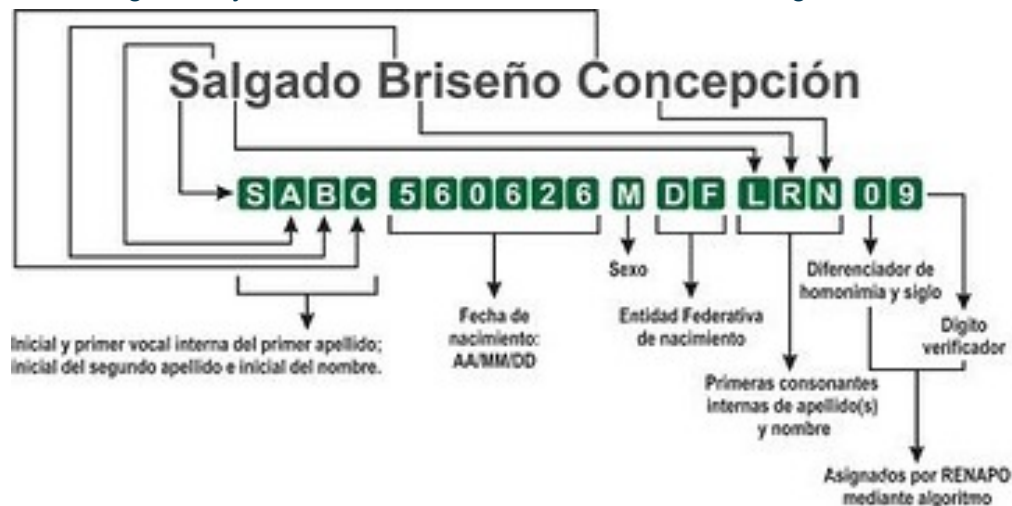
Sample Hard Declined Response

You receive the hard declined error for any integration error. HTTP status is not 2xx in this case.

```
{  
  "requestId": "c5f2b12f-5a99-4a0c-912a-aaa48abd9dc6",  
  "errorCode": "validation_error",  
  "message": "Duplicated ID: cs_a97d0c8b6cd34a94ae066ac78c571edc"  
}
```

CURP and RFC format

CURP is generally for consumers and is 18 characters in length.



RFC for companies is 12 characters and for consumers it is 13 characters.

