



Our API

Our RESTful API is a system that allows one program to let other programs talk to it directly. Just like humans interact with programs through a user interface, a program interface interacts with other programs, allowing data to pass between them. (Despite any differences in hardware or software).

API Use

When working on integrations, we recommend that you use our dev server before running your code against production. This won't create real data on our end. It's sole purpose is to test your code and confirm that it's working properly and as expected.

Development URL:

```
http://api.sandbox.facti.mx
```

Production URL:

```
https://api.facti.mx
```

Auth Credentials

You must obtain your client ID and API key in order to authenticate with our API. These can be obtained through [Facti's platform](https://www.facti.mx/api/keys) [https://www.facti.mx/api/keys]. We strongly suggest to keep these credentials private and stored in a secure place. These credentials are used to authenticate to both our development and production server.

Credentials example:

```
Client ID: d09168b32f80ai8ba24c416916y304n8  
API KEY: 26b413ah0ba67543c7642069a91h743t
```

API Response

Our RESTful API will always return a JSON object.

For successful API calls, our JSON response object looks like:

```
{  
  result: 1,  
  msg: '',  
  ...DATA_PAYLOAD  
}
```

For unsuccessful API calls, our JSON response object looks like:

```
{  
  result: 0,  
  msg: 'ERROR_MESSAGE',  
}
```

REQUEST ROUTES

1. Payment requests

HTTP Requests

POST https://api.facti.mx/v1/payments/new_request

Parameter	Required	Description
client_id	Yes	Client ID
api_key	Yes	API key
serie	No	Invoice's series.
folio	Yes	Invoice's folio.
from_rfc	Yes	RFC issued by invoice.
to_rfc	No	RFC recipient.
to_client_name	Yes	Name of the company / person to whom the payment is directed.
status	No	Invoice can be saved as "paid" to store for informational purposes. Values are 0 (pending) y 1 (paid). Default value is 0 .
payment_date	No	Date on which the payment was reconciled if the invoice was generated as paid. Format YYYY-MM-DD . In case of not sending a value, it takes the date on which the request was made.
to_email	Yes	Email of the recipient to whom the invoice is sent. It can be more than one, separated by a comma ',' .
total_amount	Yes	Invoice's total amount. Maximum of 2 decimal places, otherwise it'll be rounded up 242.831 -> 242.84
limit_payment_date	No	Payment deadline in format YYYY-MM-DD . If not specified, a month from current day will be used.
send_email_to_client	No	Indicates whether an email will be sent immediately to the customer with the payment methods so that he can make the transfer or deposit. Its value can be 0 or 1 . Default value is 1 .
payment_ways_ids	No	Array with the payment methods' IDs to be used for the payment request. Through these IDs you can limit a user to use 1 or more specific payment methods configured for a company.
folios	No	Array with invoices' folios that want to be grouped. If it's included in the request, then folio y se debe de mandar el valor fijo 1 must be sent in its place.
series	No	Array with the corresponding series to the folios sent in the previous parameter. If an invoice doesn't have a series, you must send an empty field "" in its corresponding index.

total_amounts	No	Array with the corresponding amounts of each grouped invoice. The amounts' sum must be exactly the same as the value sent in total_amount parameter.
concepts	No	Array with the concepts for the current payment. Each concept must contain the following properties: <pre>\$concepts => array(array('quantity' (required) => 2, 'unit' (requerido) => 'PIECE', 'description' (requerido) => 'Monitor', 'pu' (requerido) => 2500.00, 'total' (opcional) => 5000.00, 'details' (opcional) => '25 inch Monitor'))</pre>
generate_invoice	No	Generate invoice with payment request and attach it to the email that the client will receive. Its value can be 0 (no) or 1 (yes). Default value is 0
invoice_receiver	No	Tax information required by the client in order to generate the invoice with the payment request. Ej. <pre>'invoice_receiver' => array('rfc' (requerido) => 'XAXX010101000', 'nombre' (opcional) => 'Client's Corporate name', 'domicilio_calle' (opcional) => 'Street', 'domicilio_no_ext' (opcional) => 'Street Number', 'domicilio_no_int' (opcional) => 'Apartment', 'domicilio_colonia' (opcional) => 'Neighborhood', 'domicilio_localidad' (opcional) => '', 'domicilio_referencia' (opcional) => '', 'domicilio_municipio' (opcional) => 'District', 'domicilio_estado' (opcional) => 'State', 'domicilio_pais' (opcional) => 'Country', 'domicilio_cp' (opcional) => 'ZIP Code')</pre>

Respuesta HTTP

Parámetro	Descripción
status	Request's status. Its values can be 0 (error) and 1 (success).
msg	Error message in case of an unsuccessful request.
request_id	Payment's request unique ID.
total_amount	Total amount to pay.
limit_payment	Payment's limit date.
references	<p>Array that contains the bank references depending on the configured payment methods.</p> <p>payment_way_id => Unique ID of the payment method configured in the system (useful to limit a user's payment method).</p> <p>payment_way_alias => Internally assigned alias to the payment method configured for the company.</p> <p>payment_type => Payment type (paypal, stp, bank).</p> <p>reference => The capture line in case that 'payment_type' is 'bank' and numeric reference in case of 'stp' (assigned in bank portal).</p> <p>stp_cuenta_clabe => Only applies for 'stp' payments. Contains the CLABE account to configure in the bank portal.</p> <p>bank_name => Only applies for 'bank' payments. Indicates the bank's name for the current reference.</p> <p>bank_agreement_number => Only applies for 'bank' payments. Indicates the agreement number with the bank.</p>
References_errors	<p>Array that contains the errors (if any) obtained during the reference creation.</p> <p>type => (bank, stp o paypal) Payment's method.</p> <p>error => Error message</p>
send_to	Email address @facti.mx to which the XML and PDF of the stamped invoice must be sent.

Response example

```
{
  result: 1,
  msg: "",
  request_id: 38,
  total_amount: 55.20,
  limit_payment_date: "2013-11-21",
  send_to: "CT01109078C1@facti.mx",
  references: [
    {
      payment_way_id: "1",
      payment_way_alias: "STP Prepago",
      payment_type: "stp",
      reference: "38",
      stp_cuenta_clabe: "646180109301000010"
    },
    {
      payment_way_id: "9",
      payment_way_alias: "Santander Prepago",
      payment_type: "bank",
      reference: "380010000000096306229",
      bank_name: "Santander",
      bank_agreement_number: "4381"
    }
  ]
}
```

2. Obtain CLABE accounts for STP payments.

Through this service it's possible to obtain the CLABE accounts of a company's client. These must be registered in their bank portal for future transactions (unique company-client combination).

HTTP request

```
POST https://api.facti.mx/v1/payments/stp_client_clabe
```

Parameter	Required	Description
client_id	Si	Client ID.
api_key	Si	API key.
company_id	Si	Company's ID from which the STP CLABE accounts for the customer will be obtained.
to_email	Si	Email of the client/company from which the CLABE account must be obtained to register in the bank portal.
payment_ways_id	No	Array with the payment method's IDs to be used for the creation of the CLABE accounts. This way you can limit a user to use 1 or more specific payment methods configured by the company.

HTTP Response

Parameter	Description
status	Request's status. Its values can be 0 (error) or 1 (success)
msg	Error message in case of an unsuccessful request.
request_id	Payment request unique ID.
accounts	Array that contains the CLABE accounts for the user. Ej. Array(payment_way_id => Unique ID of thr payment method configured in the system. payment_way_alias => Internally assigned alias to payment method configured for the company. stp_cuenta_clabe => CLABE account for the user to register in its bank portal for future transactions.)
errors	Array that contains the errors (if any) obtained during the CLABE account creation.

Response example

```
{
  result: 1,
  msg: "",
  accounts: [
    {
      payment_way_id: 1,
      payment_way_alias: "STP Prepago",
      stp_cuenta_clabe: "646180109301000010"
    },
    {
      payment_way_id: 2,
      payment_way_alias: "STP Postpago",
      stp_cuenta_clabe: "646180109302000019"
    }
  ],
  errors: []
}
```

3. Company's payments search

Make inquiries about the requested payments made through the service "Payment requests". Through this endpoint, you can obtain those payments that have not yet been liquidated and those that have been reconciled/liquidated.

HTTP Request (Production)

```
POST https://api.facti.mx/v1/payments/company_payments
```

Parameter	Required	Description
client_id	Yes	Client ID.
api_key	Yes	API key.
company_id	Yes	Company's ID from which the payments will be obtained.
since_id	No	Return payments with ID from equal or greater than the one specified.
limit	No	Maximum amount of payments to be obtained. Maximum value is 200 .
max_id	No	Return payments with ID equal or less than the one specified.
status	No	Get payments with specific status, where 0 means not paid and 1 means paid/liquidated.
from_payment_date	No	Minimum date on which the payment was reconciled. It must have the following format Y-m-d H:i:s (ej. 2019-05-10 16:55:51).
to_payment_date	No	Maximum date on which the payment was reconciled. It must have the following format Y-m-d H:i:s (ej. 2019-05-10 16:55:51).

HTTP Response

Parameter	Description
status	Request's status. Its values can be 0 (error) or 1 (success).
msg	Error message in case of an unsuccessful request.
payments	Array with the queried payments. Array(id , => Payment's ID. company_id , => Company's ID. request_datetime , => Request date and time. serie , => Serie of the invoice/payment. folio , => Folio of the invoice/payment. to_rfc , => RFC of the person/company to which the payment was required (customer). to_email , => Email of the person/company to which the payment was required (customer).

total_amount, => Total amount requested.

amount_received, => Amount received by the client. By default, the system is configured to only accept payments with the exact same requested amount. Nonetheless, it has the option to accept payments with greater amount, so this parameter is also returned.

limit_payment_date, => Payment's limit date.

status, => 0 (Not paid), 1 (Paid/Liquidated).

confirmed_from, => (bank, stp, paypal) Payment's method used by the client.

confirmed_date, => Payment's date in format Y-m-d H:i:s, ej. 2019-05-10 16:55:51.

)

Ejemplo de respuesta

```
{
  result: 1,
  msg: '',
  payments: [
    {
      id: 9,
      company_id: 5,
      request_datetime: '2019-05-10 15:06:05',
      serie: '',
      folio: 3,
      to_rfc: 'CGM011210S72',
      to_email: 'alejandro@mail.com',
      total_amount: 111.11,
      amount_received: 111.11,
      limit_payment_date: '2019-06-10 00:00:00',
      status: 1,
      confirmed_from: 'stp',
      confirmed_date: '2019-05-10 16:55:51',
      no_cliente: 11,
      no_factura: 5
    },
    {
      id: 10,
      company_id: 5,
      request_datetime: '2019-05-10 15:08:02',
      serie: '',
      folio: 3,
      to_rfc: 'CGM011210S72',
      to_email: 'alejandro@mail.com',
      total_amount: 221.09,
      amount_received: 221.09,
      limit_payment_date: '2019-06-10 00:00:00',
      status: 1,
      confirmed_from: 'stp',
      confirmed_date: '2019-05-10 16:55:51',
      no_cliente: 22,
      no_factura: 6
    }
  ]
}
```

4. Payment notification (IPN)

Each time a new payment is reconciled, a real-time notification will be sent to a service (webhook) through a **POST** request.

HTTP Request (Production)

```
POST http://<DOMINIO>/<ENDPOINT>
```

Parameter	Description
ipn_password	Password assigned to the company, in format MD5 .
ipn_type	Notification type. Has a value of 1 if it's a new reconciled payment.
data	<p>New payment's information in JSON format.</p> <p>Ej.</p> <pre>{ id, => Payment's ID. company_id, => Company's ID. company_razon_social, => Corporate name of the company to which the payment was made. company_alias, => Internal alias of the Company to which the payment was made. request_datetime, => Payment's request date. serie, => Serie of the invoice linked to the payment (not applicable for prepayment). folio, => Folio of the invoice linked to the payment (not applicable for prepayment) to_email, => User's email that made the payment. total_amount, => Total amount requested by the company (2 decimals). amount_received, => Amount funded/transferred by the user (2 decimals). limit_payment_date, => Payment's limit date. confirmed_from, => Method from which the payment was reconciled ('bank', 'stp' o 'paypal'). confirmed_date, => Funded/transferred date. payment_way_id, => Payment method's ID configured by the user. payment_way_alias, => Payment method's alias configured by the user. }</pre>

5. Delete payment requests

Service to delete existing payment requests. It's important to notice that only those payments with status of '0' can be deleted. Reconciled/liquidated payments can't be deleted.

HTTP Request (Sandbox)

```
POST http://api.sandbox.facti.mx/v1/payments/delete
```

HTTP Request (Production)

```
POST https://api.facti.mx/v1/payments/delete
```

Parameter	Required	Description
client_id	Yes	Client ID
api_key	Yes	API key
company_id	Yes	Company ID from which the payments will be obtained.
id	Yes	Payment ID to delete.

Response example:

```
{
  result: 1,
  msg: "",
  payment_details: {
    id: 648124,
    empresas_id: 452,
    organizaciones_id: 232,
    request_datetime: "2019-05-22 14:37:19",
    serie: "",
    folio: 3,
    to_rfc: "GGM011210S72",
    to_email: "usuario@dominio.com",
    total_amount: 5001.50,
    status: "0",
    group_parent: "0",
    group_child: "0",
    count_childs: 0,
    childs: []
  }
}
```

6. Get payment details

Service to get payment's details by its internal ID generated by Facti while creating a payment request.

HTTP Request (Sandbox)

```
GET http://api.sandbox.facti.mx/v1/payments/(payment id)
```

HTTP Request (Production)

```
GET https://api.facti.mx/v1/payments/(payment id)
```

Example:

```
GET https://api.facti.mx/v1/payments/5022
```

Parameter	Required	Description
client_id	Yes	Client ID
api_key	Yes	API key

Response example:

```
{
  result: 1,
  msg: "",
  details: {
    result: 1,
    msg: "",
    id: 5022,
    empresas_id: 452,
    organizaciones_id: 232,
    request_datetime: "2019-05-22 14:37:19",
    serie: "A",
    folio: 6,
    to_rfc: "XAXX010101000",
    to_email: "usuario@dominio.com",
    total_amount: 1000,
    status: "1",
    group_parent: "0",
    group_child: "0",
    count_childs: 0,
    childs: []
  }
}
```

7. Change status of pending payment to “paid”.

Service to change the status of a payment, from pending to reconciled/paid. Once the payment is manually set as reconciled through the API, our system will send an IPN notification if configured.

It's important to notice that you can't change status from paid to pending.

HTTP Request (Sandbox)

```
POST http://api.sandbox.facti.mx/v1/payments/(payment id)
```

HTTP Request (Production)

```
POST https://api.facti.mx/v1/payments/(payment id)
```

Example:

```
POST https://api.facti.mx/v1/payments/5022
```

Parameter	Required	Descripción
client_id	Si	Client ID
api_key	Si	API key
payed	Si	(Possibles values: 1 , true , yes) If the parameter is sent with any of the listed values, payment will be marked as manually updated from pending to paid.

Response example:

```
{
  result: 1,
  msg: "",
  details: {
    result: 1,
    msg: "",
    id: 17542,
    empresas_id: 453,
    organizaciones_id: 221,
    request_datetime: "2019-05-22 14:25:19",
    serie: "A",
    folio: 8,
    to_rfc: "XAXX010101000",
    to_email: "usuario@dominio.com",
    total_amount: 3500,
    status: "1",
    group_parent: "0",
    group_child: "0",
    count_childs: 0,
    childs: []
  }
}
```