

Crear un pago


Acción

POST

https://api.paypetic.com/pagos

Parámetros del mensaje

Nombre	Tipo	Formato	Descripción
type	String		Tipo de pago. Valores: <ul style="list-style-type: none">• debit: Debito en cuenta o tarjeta.• online: Transacción online con tarjeta.• transfer: Transferencia en cuenta.• debin: Transacción debin• coupon: Cupon de pago. Si no se envía el tipo de transacción solo se registrara la operación y retornara en el parámetro form_url la url del checkout de pago.
collector_id	String		Identificador del comercio.
return_url	String		URL a la cual redirige el formulario de ingreso de datos de métodos de pago una vez finalizado el pago. Este servicio, también realiza un POST con información básica del pago. (ID, final_amount, status, type, collector, external_transaction_id, currency, status_detail, request_date, metadata, rejected_date, Due_Date)
back_url	String		URL a la cual redirige el formulario de pago al realizar clic en el botón regresar.
notification_url	String		URL a la cual se estarán enviado las notificaciones del pago.
external_transaction_id	String		Id externo de identificación de la transacción. Este campo no puede repetirse, debe ser ÚNICO, generado por la entidad. REQUERIDO
details	Array (Object)		Listado de pagos que conforman la transacción. Puede tener mas de un pago dentro de una transacción o referenciar a pagos pendientes ya registrados realizados. También se puede enviar pagos a diferentes comercios los cuales deben declarar y registrar la relación de los mismos. REQUERIDO
payment_id	String		Identificador del pago. Se puede enviar el identificador para obtener los datos de un pago pendiente ya registrado en la plataforma, de esta forma la transacción actual realizará el pago del mismo.
external_reference	String		Referencia o id externo a la plataforma para uso de la entidad.
concept_id	String		Identificador del concepto del pago definido por la entidad.
concept_description	String		Descripción del concepto del pago definido por la entidad.
amount	Float	##.00	Monto del pago.
collector_id	String		Identificador de la entidad. Se puede enviar un identificador distinto al de la entidad que está realizando la transacción, los cuales deben declarar y registrar la relación de los mismos.
rate	Float	##.00	Tasa de interes anual a cobrar pasada la fecha de vencimiento.
charge_delay	Float	##.00	Monto fijo a cobrar pasada la fecha de vencimiento.
currency_id	String		Identificador de moneda. Consulte aquí los valores posibles. REQUERIDO
payment_methods	Array (Object)		Listado de medios de pago.
authorization_transaction_id	String		ID de transacción de autorización.
amount	Float		Monto a pagar con el medio de pago.
media_payment_id	Integer		Identificador del medio de pago.

number	String		Número del medio de pago.
installments	Integer		Cuotas.
promotion_id	String		Identificador de la promoción.
expiration_year	Integer		Año de expiración de la tarjeta
expiration_month	Integer		Mes de expiración de la tarjeta
security_code	String		Codigo de seguridad de la tarjeta.
holder	Object		Titular del medio de pago.
name	String		Nombre.
identification	Object		Documento del titular.
type	String		Tipo de documento.
number	String		Número.
country	String	ISO 3166-1 alfa-3	País de expedicion.
payer	Object		Datos del pagador. 
id	String		Identificador del pagador. Se obtiene del servicio de CUSTOMERS
external_reference	String		Referencia del pagador en la entidad.
name	String		Nombre.
email	String		Email.
identification	Object		Documento del pagador.
type	String		Tipo de documento.
number	String		Número.
country	String	ISO 3166-1 alfa-3	País de expedicion.
phones	Array (Object)		Lisado de números telefónicos.
description	String		Descipción del número.
country_code	Integer		Código de país.
area_code	Integer		Código de área.
number	Integer		Números telefónico.
extension	Integer		Extensión.
due_date	Date	yyyy-MM-dd'T'HH:mm:ssZ RFC 822 time zone	Fecha de vencimiento.
last_due_date	Date	yyyy-MM-dd'T'HH:mm:ssZ RFC 822 time zone	Fecha de último vencimiento.
metadata	Object		Objeto JSON para adjuntar datos al pago.
carrier	String		Carrier
presets	Object		Objeto JSON para presetear valores de un pago.
media_payment_ids	Array [Integer]		Lista de medios de pago aceptados por el pago

type	String		Tipo de forma de pago. Valores: <ul style="list-style-type: none"> • debit: Debito en cuenta o tarjeta. • online: Transacción online con tarjeta. • transfer: Transferencia en cuenta. • debin: Transacción debin • coupon: Cupon de pago.
promotion_ids	String		Promoción preseteada para el pago
installments	Integer		Cuotas preseteadas para el pago
actions	Array [Object]		Lista de acciones presetadas para el pago. Valores: <ul style="list-style-type: none"> • retry: true false

Guía de Uso

• Solicitar autorización

Para consumir el servicio detallado en esta sección debe solicitar una autorización. [En este link](#) se explica en detalle como obtenerla.

• Qué es una solicitud de pago?

Es el conjunto de datos que se envían y reciben cuando se trabaja con nuestra API de pagos.

Creamos una cuando queremos que un usuario pague a través de Pay per TIC.

Todos nuestros servicios son REST, envían y reciben la información en [formato JSON](#), es muy importante enviar el encabezado correspondiente en todas las llamadas a servicios de Pay per TIC, de la siguiente manera:

```
curl -X POST https://api.paypertic.com/pagos -H 'Content-Type: application/json'
```

• Qué información contiene una solicitud de pago?

Las solicitudes contienen toda la información referente a una transacción en particular: el usuario al que corresponde el pago, el importe, concepto, forma de pago, estado, etc.

• Cómo se crea una solicitud de pago?

Para crear una solicitud de pago **pendiente**, el primer paso es armar nuestro objeto JSON, luego enviarlo al [servicio web de creación de pagos](#). Mostraremos los campos mínimos y más importantes que pueden utilizarse en la creación de una solicitud de pago.

Campos requeridos:

external_transaction_id: Uno de los campos más importantes, un identificador generado por la entidad que represente cada solicitud de pago. Este campo, de tipo String (Cadena de caracteres), es requerido y debe ser único. En el caso de intentar crear dos solicitudes de pago con el mismo valor, la segunda fallará en esta validación. Esto garantiza que no se informe la misma solicitud más de una vez.

currency_id: Moneda del pago, representación en el código ISO 4217. Por ejemplo, para Pesos Argentinos: ARS.

details: Aquí van todos los conceptos de transacción. El campo es un listado de tipo Array, esto nos permitirá enviar más de un detalle en cada solicitud, nos servirá para mostrar los conceptos de forma desagregada en el caso de que se estén pagando más de uno. Cada concepto tendrá su propio importe pero la moneda debe ser la misma, ya que se indica en un nivel superior. La API sumará los importes de todos los detalles indicados.

due_date: Fecha de vencimiento. **Es la fecha predeterminada de cobro para los débitos y cupones de pago.** Por ejemplo, la entidad informa una solicitud de pago el día 3 de Mayo de 2019 con fecha de vencimiento el 30 del mismo mes. Si el usuario ingresa al formulario, selecciona pagar con un débito, ya sea en un CBU o una tarjeta y completa los datos de su medio de pago el día 5 también de ese mes, el mismo quedará en estado emitido (issued) hasta que se acerque la fecha de vencimiento, es muy importante tener esto en cuenta al momento de determinar esta fecha, normalmente, para cobranzas mensuales, la misma varía entre el 10 y el 15 del mes. **En el caso de pagos online no es relevante ya que la transacción se procesa en el momento.** Todas las fechas deben enviarse de acuerdo a la norma ISO 8601 combinando la Representación Completa en su Formato Extendido y la Hora Local Relativa, por ejemplo: "2019-12-01T00:00:00-0300". Si bien no es obligatorio enviar este campo, todas las solicitudes de pago tienen una fecha de vencimiento predeterminada. Si no se envía, será asignada por la API. Recomendamos que la envíen para que la misma esté alineada con la lógica de negocio de la institución. La comparación que se hace para determinar si una solicitud está vencida es "menor que", no inclusive a la fecha actual.

last_due_date: Última fecha de vencimiento. Una vez pasada esta fecha, ya no se podrá pagar esta solicitud. Es muy importante tener en cuenta esta fecha, ya que que determina [hasta cuándo estará disponible un formulario de pago](#). Todas las solicitudes tienen una fecha de último vencimiento y no es obligatorio enviarla, al igual que la fecha de vencimiento, si no lo hacemos se asigna una automáticamente. Recomendamos enviar una fecha que se ajuste a las necesidades para evitar inconvenientes como pago fuera de término o formularios vencidos antes de tiempo. La comparación que se hace es "menor que" no inclusive, al igual que la fecha de vencimiento. Ejemplo: Si queremos que un pago se pueda pagar hasta el último momento del año 2019 hora Argentina basta con enviar "2020-01-01T00:00:00-0300".

Cada detalle debe contener los siguientes valores:

amount: Importe del concepto o detalle. Debe estar expresado en números con punto como separador de decimales (.) y sin separador de miles, por ejemplo: 15181.85.

concept_id: Identificador del concepto. Este campo identifica el concepto que se está pagando, por ejemplo una patente, número de inmueble, de tasa o un código de producto. El valor es de tipo String y no se realiza ninguna validación sobre él. Debe ser asignado por la entidad. Recomendamos que sea un valor realmente representativo del concepto que se está cobrando, para luego ser utilizado como filtro de diferentes consultas.

concept_description: Descripción del concepto, campo de tipo String. Debemos ingresar un texto descriptivo, breve y conciso, para que el pagador lo reconozca, por ejemplo: "Cuota social Abril 2010" o "Impuesto patente ABC123 - Junio 2015".

external_reference: Referencia del pago. Aquí debemos enviar un identificador que represente a este detalle en particular, por ejemplo, un número de factura, cedulón, o simplemente un código. La función es similar al `external_transaction_id` salvo que aquí no se valida la unicidad del valor, el mismo `external_reference` puede estar presente en diferentes detalles.

Campos no requeridos, pero muy importantes:

payer: Pagador. Este campo es un objeto complejo que representa al usuario final, es decir al pagador al que le corresponde esta solicitud de pago.

Debemos enviar los siguientes valores:

name: Nombre del pagador. Campo requerido, de tipo String, debe contener el apellido y nombre del usuario final, de preferencia en dicho orden y sin separadores, o razón social, por ejemplo: "Lopez Pablo Hernan", o "EMPRESA DEMO SA". Tener en cuenta que cualquier carácter no sea una letra o espacio será eliminado.

email: Dirección de correo electrónico. No es obligatorio enviarlo para solicitudes de pago pendientes, representa el correo electrónico del usuario y se valida que el formato del mismo sea correcto, por ejemplo: "casilla@paypertic.com".

identification: Identificación del pagador. Un objeto complejo que representa algún documento oficial de identificación del pagador, como por ejemplo un DNI o un CUIT. No es requerido, pero recomendamos enviarlo en caso de conocer la información. Tiene la siguiente estructura:

type: Tipo de identificación. String que indica el tipo de identificación que estamos enviando, puede tomar dos valores:

DNI_ARG: Para DNI emitidos en Argentina.

CUIT_ARG: Para CUIT/CUIL de Argentina.

number: Número de identificación. Es el número de DNI o CUIT sin espacios ni guiones, por ejemplo: "33888444" o "30772229991".

country: País que expide la identificación representado con el código alfa-3 de la norma ISO 3166-1, por ejemplo: "ARG".

external_reference: Referencia del pagador. Este campo de tipo String no es requerido, pero en caso de enviarlo, es importante que sea un número o código que represente al usuario final en la institución ya que puede ser necesario para determinadas funcionalidades de la plataforma. En caso de no contar con un identificador para los pagadores, recomendamos utilizar el DNI/CUIT o generar uno.

notification_url: URL de notificaciones. URL en la que se encuentra el servicio web que recibirá las notificaciones sobre esta transacción. No es requerido, pero es **la mejor forma de lograr una integración 100% automatizada y en tiempo real**, por lo que es sumamente recomendado. Más información sobre este punto en el apartado correspondiente, servicio de notificaciones.

Con estos simples campos ya tenemos la base necesaria para cualquier solicitud de pago, un ejemplo debería verse así:

```
{
  "external_transaction_id": "1238219",
  "currency_id": "ARS",
  "details": [{
    "external_reference": "1123",
    "concept_id": "ABC123",
    "concept_description": "Pago de ABC123 - Junio 2021",
    "amount": 3804.50
  }],
  "payer": {
    "external_reference": "2932"
    "name": "Usuario final",
    "email": "pagador@paypertic.com",
    "identification": {
      "type": "DNI_ARG",
      "number": "11222333",
      "country": "ARG"
    }
  },
  "due_date": "2020-12-10T00:00:00-0300",
  "last_due_date": "2020-12-24T00:00:00-0300",
  "notification_url": "https://host.com/notificaciones"
}
```

Ahora que tenemos nuestro objeto podemos preparar el mensaje completo: Debe ser de tipo POST y enviado a <https://api.paypertic.com/pagos>.

Recordamos que debemos haber **solicitado autorización** y el formato del cuerpo del mensaje debe ser JSON, por lo que tendremos dos encabezados. El mensaje quedaría de la siguiente manera:

```
curl -X POST https://api.paypertic.com/pagos -H 'Authorization: Bearer token' -H 'Content-Type: application/json' -d '{
  "external_transaction_id": "1238219",
  "currency_id": "ARS",
  "details": [{
    "external_reference": "1123",
    "concept_id": "ABC123",
    "concept_description": "Pago de ABC123 - Junio 2021",
    "amount": 3804.50
  }],
  "payer": {
    "external_reference": "2932"
    "name": "Usuario final",
    "email": "pagador@paypertic.com",
    "identification": {
      "type": "DNI_ARG",
      "number": "11222333",
      "country": "ARG"
    }
  },
  "due_date": "2020-12-10T00:00:00-0300",
  "last_due_date": "2020-12-24T00:00:00-0300",
  "notification_url": "https://host.com/notificaciones"
}'
```

Una vez que enviamos el POST, si el mensaje fue correcto, **recibiremos como respuesta un código 200** que indica que Pay per TIC registró la transacción correctamente. Si no recibimos un 200 debemos verificar los pasos anteriores o ver los posibles errores descritos [en esta página](#). La respuesta tendrá campos nuevos, analizaremos los más importantes:

id: Identificador de la transacción de Pay per TIC. La API asigna un id de tipo uuid a todas las transacciones que registra. Este campo es muy importante, recomendamos vincular este id en vuestro sistema con el external_transaction_id informado en la solicitud, puesto que ambos id son únicos. Este identificador de transacción permitirá consultar los servicios de Pay per TIC de forma más eficiente.

form_url: URL del formulario de pago. Es el link que debemos brindar al usuario para que realice el pago. Este link mostrará el estado de la transacción en todo momento, en el caso de un pago pendiente dará las opciones de pago disponibles, si el mismo ya se encuentra pago, mostrará un comprobante, o el motivo de rechazo en caso de un pago no exitoso, etc.

final_amount: Monto total de la transacción. Es la suma de todos los detalles de la solicitud más los costos adicionales que pudiera haber (como traslado de costos y aranceles, o costo de financiación por pago en cuotas.)

status: Estado de la solicitud. Este campo es de suma importancia, indica en qué estado se encuentra la solicitud, aquí encontrará un [listado completo de estados de una solicitud de pago](#).

last_update_date: Fecha de última actualización. Indica cuándo se realizó el último cambio sobre esta solicitud en nuestra plataforma.

En este punto resta que el usuario ingrese al formulario, seleccione una forma de pago y complete los datos.

• Cómo podemos probar estos formularios?

Para verificar el funcionamiento de los formularios disponemos de [Tarjetas de prueba](#) que nos servirán para realizar pagos online aprobados y rechazados.

Estas tarjetas **sólo sirven en modo STARTING** o entorno de pruebas de la plataforma. Todas las entidades se encuentran en este entorno al comienzo, salvo que ya hayan solicitado el <link>pase a producción</link>.

Si ingresamos al formulario de pago, completamos los datos y pagamos con una de estas tarjetas, dependiendo de la que hayamos utilizado, se mostrará el comprobante de pago, o un formulario indicando el motivo de rechazo y ofreciendo la posibilidad de reintentar.

En ese preciso instante deberíamos recibir una notificación en el servicio web enviado en notification_url indicando que la solicitud de pago cambió de estado y brindando todos los detalles para impactarlos en vuestro sistema.

Recordamos que la notificación se envía con todos los datos disponibles de la solicitud de pago, por lo tanto el servicio debe estar preparado para recibir todos los campos indicados en la documentación, recomendamos ignorar los campos desconocidos para evitar problemas de compatibilidad con nuevas versiones de la API.

De esta forma concluye el ciclo básico de cualquier solicitud de pago de forma espontánea:

1. Solicitar autorización
2. Crear la solicitud de pago
3. Mostrar el formulario
4. Recibir la respuesta.

Esta integración permitirá generar todo tipo de solicitudes y obtener en tiempo real las actualizaciones sobre los pagos de forma automática, la cantidad de veces que sea necesario.

Recomendamos de sobremanera leer el [listado completo de servicios web](#) para conocer todos los end points a disposición.

Algunas configuraciones adicionales para esta integración son: [configuración de emails](#), <link>personalización de formularios y templates</link>, <link>configuración de medios de pago, estrategia y promociones</link>

Errores Posibles