



## **FASTPAY**

Especificaciones de Integración Vía API



03/03/2022

V 2.1

**DOCUMENTO CONFIDENCIAL**

## Contenido del documento

1. INTRODUCCIÓN .....	2
2. ALTA COMERCIO Y OBTENCIÓN DE CREDENCIALES (API).....	4
3. URLS ENTORNOS .....	4
4. CONCEPTOS PARA LA INTEGRACIÓN DEL PRODUCTO.....	5
5. INTEGRAR EL CONFIGURADOR DE FASTPAY (BACKOFFICE) .....	13
6. COMPATIBILIDAD CON OTROS NAVEGADORES.....	13
7. FORMATO DE CAMPOS .....	14
8. PSD2 .....	17
9. CONSIDERACIONES PARA LA ACTIVACIÓN EN PRODUCCIÓN (LIVE) .....	22

## 1. INTRODUCCIÓN

Imagine por un momento que cuando va a comprar su comida favorita pudiera pagar de la manera más rápida, cómoda y segura porque tiene a su disposición los métodos de pago más utilizados. ¿Y si pudiera hacerlo desde cualquier dispositivo? ¿Y si el pago fuera el dispositivo? ¿Y si pago con el método que siempre utilizo porque es lo más sencillo? Esto me permitiría empezar a disfrutar antes de mi compra. Y volvería, porque mi experiencia ha sido cómoda y flexible. Porque yo elijo pago.

Piense como comercio lo que podría ser para sus clientes que tuvieran acceso a todas las formas de pago disponibles en el mercado a través de una sencilla integración de un solo producto. Ya no tendría que afrontar costosas integraciones cada vez que quiera añadir más formas de pago, cambiar el orden, decidir cuándo mostrarlos, etc.

Si hay algo que un comercio sabe es que sus clientes son más exigentes cada día. Van a querer seguir utilizando las formas de pago más utilizadas de la actualidad y aquellas que están por venir. Eso es conversión, la compra más cómoda y rápida. Sin fricción y con total seguridad.

Todo esto es FastPay.

The image shows two parts of the FastPay interface. The top part is a menu of payment options under the heading "Pago con tarjeta". The options listed are: bizum, PayPal, G Pay, Alipay, WeChat Pay, and TRANSFERENCIA BANCARIA PSD2. The bottom part is a card payment form. It includes a sidebar with icons for the same payment methods. The main form has the following fields and elements:
 

- Field: "NOMBRE EN LA TARJETA:"
- Field: "INFORMACIÓN DE LA TARJETA:"
- Fields: "MM / YY" and "CVV" (with a help icon).
- Checkbox: "Recordar los datos de mi tarjeta para futuras compras. más info."
- Button: "PAGAR 93,50 €"
- Security logos: "Pago seguro", "MasterCard SecureCode", and "Verified by VISA".

El siguiente documento detalla los requerimientos y funcionalidades necesarios para poder llevar a cabo la **integración vía API de FASTPAY de SIPAY** en tu comercio online.

**FastPay** es una solución que te permite disponer de una gran variedad de métodos de pago de una forma rápida y segura, mejorando los ratios de conversión de tu tienda online, ofreciendo a tus clientes la posibilidad de pagar cómo, cuándo y dónde quieran.

Mediante la integración del FastPay, podrás añadir en tu tienda online las siguientes formas de pago para tus clientes:

- Pagos con tarjeta.
- Tokenización de tarjeta (para compras recurrentes)
- Pagos alternativos:
  - PayPal
  - Bizum
  - Google Pay
  - Apple Pay
  - WeChat Pay
  - Alipay
  - Amazon Pay
  - Transferencias inmediatas

El FastPay está respaldado por la seguridad que ofrece la certificación **PCI DSS 3.2.1 Level 1**, la normativa internacional de seguridad para aquellas entidades que almacenan, procesan o transmiten datos de titulares de tarjeta o datos sensibles de autenticación, gracias a la tecnología basada en iframe con el cifrado y **tokenización TLS 1.2**.

Todas las llamadas enviadas al API de SIPAY deberán ser firmadas con el algoritmo HMAC-SHA256, teniendo en cuenta que, a la hora de realizar la firma, los datos deben estar previamente deserializados dado que JSON no respeta el orden. Una vez calculada la firma en formato hexadecimal, deberemos enviar su valor en una cabecera http denominada Content-Signatura además del JSON con los datos de la petición.

## 2. ALTA COMERCIO Y OBTENCIÓN DE CREDENCIALES (API)

Una vez esté decidido a integrar el producto de FastPay de Sipay, se le asignará un service manager para que le ayude en todo el proceso de integración.

Su service manager será persona encargada de configurar y darle de alta en nuestra pasarela para enviarle las credenciales de autenticación del API que serán necesaria para que funcionen las transacciones y pueda empezar a vender online.

Credenciales necesarias para sandbox y producción:

- Key
- Secret
- Resource.

## 3. URLS ENTORNOS

A continuación, se detallan las URLs/endpoints a los que se debe apuntar para hacer las peticiones API.

Para las peticiones servidor a servidor. Se reenvía la petición a nuestro endpoint firmada con HMAC-SHA256. Dependiendo del entorno, deberemos enviar la petición a un URL distinta:

- Sandbox (a.k.a Integración): <https://sandbox.sipay.es/pwall/api/v1/actions>
- Live (a.k.a Producción): <https://live.sipay.es/pwall/api/v1/actions>

Para las peticiones para el renderizado del FastPay (front), se deben utilizar estas URLs en función del entorno:

- Sandbox (a.k.a Integración)
  - [https://sandbox.sipay.es/pwall\\_app/css/app.css](https://sandbox.sipay.es/pwall_app/css/app.css)
  - [https://sandbox.sipay.es/pwall\\_sdk/pwall\\_sdk.bundle.js](https://sandbox.sipay.es/pwall_sdk/pwall_sdk.bundle.js)
  - [https://sandbox.sipay.es/pwall\\_app/js/app.js](https://sandbox.sipay.es/pwall_app/js/app.js)
- Live (a.k.a Producción)
  - [https://live.sipay.es/pwall\\_app/css/app.css](https://live.sipay.es/pwall_app/css/app.css)
  - [https://live.sipay.es/pwall\\_sdk/pwall\\_sdk.bundle.js](https://live.sipay.es/pwall_sdk/pwall_sdk.bundle.js)
  - [https://live.sipay.es/pwall\\_app/js/app.js](https://live.sipay.es/pwall_app/js/app.js)

#### 4. CONCEPTOS PARA LA INTEGRACIÓN DEL PRODUCTO

La integración de FastPay está simplificada para que tan solo consista en los siguientes pasos:

**[FRONTEND] Visualización del FastPay** → Consiste en incorporar la carga de la librería JavaScript en la página web del comercio. Es decir, debe incorporarlas en la página donde se realizará el checkout del pedido. Revise los ejemplos ('2\_web de ejemplo - ver codigo.html') que habrá encontrado junto a esta documentación para analizar los ejemplos entregados. El código está comentado y es muy recomendable que dedique unos minutos a examinarlo.

Esta integración consiste en los siguientes pasos:

**IMPORTANTE:** En todos los ejemplos código donde aparezca <environment> estaremos indicando que debe ser sustituido por el identificador del entorno:

- *sandbox* → Entorno de Integración
- *live* → Producción

##### 1. Importación de librería principal y su CSS

```
<link href="https://<environment>.sipay.es/pwall_app/css/app.css" rel=stylesheet>
<script src="https://<environment>.sipay.es/pwall_sdk/pwall_sdk.bundle.js"></script>
```

##### 2. Definir una capa div con un id concreto

```
<div id="app"></div>
```

##### 3. Codificar el script principal con sus parámetros

```
<script data-url="url_to_backend"
  data-amount="100"
  data-currency="EUR"
  data-placeholder="#app"
  src="https://<environment>.sipay.es/pwall_app/js/app.js">
</script>
```

Los posibles parámetros son:

Campo	Descripción	Formato	Obligatorio / Opcional
data-url	URL absoluta o relativa del backend el Comercio.	URL	Obligatorio

Campo	Descripción	Formato	Obligatorio / Opcional
data-amount	Importe de la operación.	Cadena/string. Valor entero del importe a cobrar con dos decimales en el caso del euro. Ejemplo: 10 euros serían "1000" porque completamos a 10,00.	Obligatorio
data-currency	Acrónimo ISO para indicar la moneda de cobro.	Acrónimo ISO.	Obligatorio
data-placeholder	Identificador "id" del elemento html que contendrá el renderizado del muro de pagos.	Atributo ID del elemento html.	Obligatorio.
data-catcher	<p>Indicaremos si queremos que se haga automáticamente el procesamiento del cobro con tarjeta o solo utilizaremos para capturar los datos.</p> <p>Para obtener el identificador temporal de la captura, se ha implementado un evento nuevo:</p> <p>payment_wall_catcher_data</p> <p>Ejemplo de uso:</p> <pre>window.pwall.placeholder.addEventListener('payment_wall_catcher_data', (event) =&gt; {console.log(event.detail)})</pre> <p>Dentro de event.detail estará el campo request_id con el valor que utilizaremos para el resto de llamadas server-to-server documentadas en esta referencia:</p>	<p>true: activaremos el modo capturador exclusivo.</p> <p>false: activaremos el modo capturador + procesador.</p>	Opcional. El valor por defecto es false.

Campo	Descripción	Formato	Obligatorio / Opcional
data-one-click-payment	<p><a href="https://sandbox.sipay.es/docm/psd2/es/#/peticiones/autenticacion">https://sandbox.sipay.es/docm/psd2/es/#/peticiones/autenticacion</a></p> <p>El contenido de catcher sería el siguiente:</p> <pre> ... "catcher": {   "type": "fpay",   "token_fpay": "valor_del_request_id_recogido_en_front", } ... </pre> <p>Después de la llamada server-to-server, y en función del resultado de la operación, deberemos lanzar los siguientes callbacks en el frontend:</p> <pre> // Procesamiento correcto window.pwall.callbacks.payment_ok({});  // Procesamiento incorrecto window.pwall.callbacks.payment_ko({}); </pre> <p>Indicaremos si queremos que los métodos de pago se activen automáticamente sin abrir el acordeón.</p>	<p>true: Apple Pay y Google Pay se activarán con un solo click. false: Todos los métodos de pago necesitarán ser utilizados con el sistema de acordeón.</p>	Opcional. El valor por defecto es false.

#### 4. Programar la escucha y lanzamiento de los eventos JavaScript

En este ejemplo se codifican la captura y lanzamiento de eventos obligatorios y en el orden de anidamiento correcto.

Se podrán realizar variaciones en el código siempre y cuando se respete los mismos nombres y orden de lanzamiento/escucha.

```

<script>
  // Manage redirect with notify.result

```





```
const urlParams = new URLSearchParams(window.location.search);
const error = urlParams.get('error');
const method = urlParams.get('method');
const request_id = urlParams.get('request_id');
if (request_id) {
  window.PaymentWall.listenTo(document, "payment_wall_setup", function () {
    var placeholder = document.getElementById('app');
    window.PaymentWall.listenTo(placeholder, 'payment_wall_drawn', function () {
      window.pwall.dispatch('process_redirect', {
        detail: {
          "error": error || null,
          "method": method,
          "request_id": request_id
        }
      });
    });
  });
}
window.PaymentWall.start()
</script>
```

##### 5. Programar la recogida de parámetros tipo QueryString

Respecto al código anterior, en este ejemplo hemos utilizado las funciones estándar de JavaScript para parsear los valores del QueryString de la URL. Se podría utilizar cualquier función propia de VueJS, React, Angular y otras librerías que permitan realizar lo mismo.

```
const urlParams = new URLSearchParams(window.location.search);
const error = urlParams.get('error');
const method = urlParams.get('method');
const request_id = urlParams.get('request_id');
```

**[BACKEND] Integrar la única llamada API** → Tan solo deberá integrar una llamada API para permitir el pago con cualquiera de los métodos de pago soportados (revise la contratación). Básicamente tendrá que desarrollar las clases, métodos y funciones necesarias para utilizar una llamada API de tipo REST con mensajería JSON. La web del comercio (frontend) enviará todos los mensajes significativos al servidor del comercio (backend) en formato JSON y solo a través de HTTPS (parámetro data-url de la librería JavaScript).

FastPay garantiza al 100% que nunca pasarán datos de tarjeta a través del servidor del comercio descargando así de las responsabilidades más estrictas de PCI DSS (SAQ-D).

Para permitir que el comercio pueda hacer pruebas, tenemos habilitado un entorno llamado sandbox que está disponible las 24 horas del día.

Con el fin de asegurar sus transacciones, todos los datos serán cifrados mediante el protocolo HTTPS TLSv1.2.

Adicionalmente, le proporcionaremos unas claves de integración por entorno. Estas claves son conocidas como: Key, Resource y Secret. Serán distintas en función del entorno (pruebas y real).

El servidor del comercio deberá codificar un programa que llamaremos **INSPECTOR** que permita revisar todas las respuestas del servidor de SIPAY. Deberá ser capaz de discriminar las respuestas que debe ignorar y enviarlas a la web del comercio (frontend) como respuestas de la petición inicial. Este proceso es conocido como **PROXY**. Principalmente su integración en el backend será un programa que hará de intermediario entre el frontend y el servidor de SIPAY.

A continuación, introduciremos las reglas que permiten interceptar la respuesta con el resultado de una operación de venta. Esta será la evidencia que permite hacer el **CIERRE DEL PEDIDO**. Normalmente el cierre del pedido lleva asociados procesos internos del comercio y ajenos a esta integración pero que son importantes para el cobro y entrega del producto al cliente.

En la programación del componente **INSPECTOR** deberemos examinar siempre el valor recibido de la web del comercio (frontend) del campo "action" y del campo "params.is\_backoffice". Si "params.is\_backoffice" tiene valor "true" deberemos descartar la petición y no enviarla al servidor de SIPAY. Se tienen que cumplir todos los criterios descritos a continuación para poder hacer el **CIERRE DEL PEDIDO** con éxito (operador lógico Y):

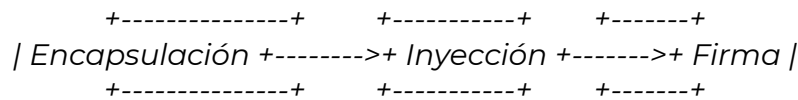
- En la petición del servidor del comercio (frontend) aparece el campo "action" con valor "pwall.sale"
- En la respuesta de SIPAY el "code" es 0 dentro de "result"
- En la respuesta de SIPAY en "result.payload" no existe el campo "url"
- En la respuesta de SIPAY, en result.payload, no hay un campo "code" con valor "198" (string) para operaciones de DCC.

En caso de que el pago ha sido fallido el comportamiento por defecto de FastPay será el de permitir un nuevo intento.

Reglas de pago fallido:

- En la petición del servidor del comercio (frontend) aparece el campo "action" con valor "pwall.sale"
- En la respuesta de SIPAY el "code" es distinto de 0 dentro de "result"

La programación en el backend deberá contemplar estos pasos:



**Encapsulación:** crearemos una estructura básica en formato JSON e introduciremos los datos recibidos de la web del comercio (frontend) en el nivel identificado como 'payload'. En este paso, no manipularemos ni modificaremos ningún valor recibido por la web del comercio, introduciendo los datos originales en la nueva estructura.

```

Estructura básica
{
  "payload": {
    // ENCAPSULACIÓN DE LOS DATOS RECIBIDOS DEL FRONTEND
  }
}
    
```

**Inyección:** consiste en añadir campos adicionales a los mensajes intercambiados entre los actores de la integración. Parte de esta mensajería es enviada a través del servidor del comercio (backend) desde la web del comercio (frontend).

## Estructura básica con campos inyectados

```

{
  "key": "<identificación del API para el comercio>", // INYECCIÓN OBLIGATORIA
  "resource": "<recurso de API para el comercio>", // INYECCIÓN OBLIGATORIA
  "nonce": "<marca de tiempo para la seguridad de la transacción>", // INYECCIÓN
OBLIGATORIA
  "mode": "sha256", // INYECCIÓN OBLIGATORIA
  "payload": {
    // DATOS ENCAPSULADOS
    ...
    params {
      ...
      "emv3ds": <parametros opcionales para psd2>,
      "reason": <razon de peticion de exencion, opcional, solo si se especifica exenciones>,
      "sca_exemptions": <exenciones a aplicar, opcional>,
      "amount": <importe de la operación>, // INYECCIÓN OBLIGATORIA
      "currency": "<ISO4217 de la moneda en formato cadena mayúsculas>", // INYECCIÓN
OBLIGATORIA
      "group_id": "0", // INYECCIÓN OBLIGATORIA, solo para tokenización de tarjetas.
      "notify": { // INYECCIÓN OBLIGATORIA
        "result": "http://localhost:63195/SIPAYResult.aspx" // INYECCIÓN OBLIGATORIA
      },
      "order": "257bdd-ea8e-417", // INYECCIÓN OPCIONAL, generaremos un order
aleatorio en caso de no ser enviado
      "token": "fca052-a60b-44b" // INYECCIÓN OPCIONAL, solo para tokenización de
tarjetas.
      ...
    }
    ...
  }
}

```

Resumen de los campos inyectables:

Obligatorios: key - resource - nonce - mode - amount - currency - group\_id - notify.result  
 Opcionales: order - token

*Al final de la documentación encontrará una sección de dedicada al formato de campos. Revisela para mayor comprensión sobre las limitaciones y para tener más ejemplos.*

**Firma digital:** como seguramente conozca, un mensaje JSON no es más que un texto con un formato determinado por un estándar. La firma digital consiste en utilizar una función programática para generar un HMAC SHA256 con su clave secreta (secret) ejecutado sobre el mensaje JSON recién 'encapsulado' e 'inyectado'.

El resultado del HMAC SHA256 es una cadena en formato hexadecimal variable en función del contenido, y de longitud fija. Deberá convertir este resultado a formato minúsculas.

Cuando se realice la petición al API de SIPAY, deberá siempre calcular la firma y enviarla en una cabecera HTTP llamada 'Content-Signature' como valor.

Deben considerarse dos premisas muy importantes:

1. El formato de envío y firma del JSON deben ser el mismo. Normalmente hay dos formatos de envío: one-line y pretty

Ejemplo JSON 'one-line':

```

{"clave1": "valor1","clave2": "valor2","nivel2": {"nivel2_clave1": "nivel2_valor1","nivel2_clave2": "nivel2_valor2"}}
    
```

Ejemplo JSON 'one-line':

```

{
  "clave1": "valor1",
  "clave2": "valor2",
  "nivel2": {
    "nivel2_clave1": "nivel2_valor1",
    "nivel2_clave2": "nivel2_valor2"
  }
}
    
```

2. Una vez realizada la firma, no se podrá modificar el formato del mensaje ni su contenido.

Hemos incluido algunos ejemplos de firma digital utilizando la herramienta <https://www.freeformatter.com/hmac-generator.html> :

Petición JSON	Secreto API (Secret Key)	Firma digital HMAC SHA256
<pre> {   "clave1": "valor1",   "clave2": "valor2",   "nivel2": {     "nivel2_clave1": "nivel2_valor1",     "nivel2_clave2": "nivel2_valor2"   } }         </pre>	ejemplo-firma	0ec34613b306bbc30b8087c5d3ecb5d95494b77d07e3071781be95a16827e0a4
<pre> {"clave1": "valor1","clave2": "valor2","nivel2": {"nivel2_clave1": "nivel2_valor1","nivel2_clave2": "nivel2_valor2"}} {   "clave1": "valor1",   "clave2": "valor2",   "nivel2": {     "nivel2_clave1": "nivel2_valor1",     "nivel2_clave2": "nivel2_valor2"   } }         </pre>	<p>ejemplo-firma</p> <p>otroEjemplo</p>	<p>94ade75e59a4139c90d307642ef1688baa9b261dc682ac8bcd5e2340312d64a0</p> <p>4796d25fa51fcd620417010c978980e64999c38382d8d73100e528067cb970f</p>



Petición JSON	Secreto API (Secret Key)	Firma digital HMAC SHA256
<pre>{   "clave1":   "valor1",   "clave2":   "valor2",   "nivel2":   {     "nivel2_clave1":     "nivel2_valor1",     "nivel2_clave2":     "nivel2_valor2"} }</pre>	otroEjemplo	bef65d9e51183426c69ebe46e628b09355edad71ab29465e9d8775febe487c3a

Para facilitar su integración hemos buscado algunas funciones para generar firmas HMAC ordenado por lenguaje de programación:

- Python: hmac, hashlib
- NodeJS: crypto
- PHP: hash\_hmac
- Java: javax.crypto.Mac, javax.crypto.spec.SecretKeySpec, org.apache.commons.codec.binary.Base64
- C#: System.Security.Cryptography
- Ruby: openssl
- Perl: Digest: SHA
- PowerShell: System.Security.Cryptography.HMACSHA256
- Shell Script: echo -n \$MESSAGE | openssl dgst -sha256 -hmac \$SECRET -binary
- Objective C and Cocoa: CommonCrypto/CommonHMAC.h
- Go: "crypto/hmac", "crypto/sha256"

Finalmente incluimos un ejemplo de firma codificado en Python:

```
import requests
import json
import hashlib
import hmac

body = {
  "key": "589365da65c48cff87d0874a",
  "nonce": "1234567890",
  "mode": "sha256",
  "resource": "359ef8ce5c5f4003b71692e446908c27",
  "payload": {
    "param1": "value1",
    "param2": "value2"
  }
}

secret = b'api-secret'
sign = hmac.new(secret, json.dumps(body).encode('utf-8'), hashlib.sha256).hexdigest()

url = 'https://dummy.es/'
response = requests(url, data=body, headers={'Content-signature': sign})
```

## 5. INTEGRAR EL CONFIGURADOR DE FASTPAY (BACKOFFICE)

La integración de este componente aprovecha todos los pasos anteriores. Este componente permite configurar el orden de los métodos de pago, la visibilidad, rótulos y títulos, etc.

Se recomiendan estas indicaciones para la correcta integración del backoffice:

- Integrarlo en una página distinta a la del pago.
- Controlar los permisos y la sesión de un usuario administrador antes de visualizar el configurador.
- Chequear la petición del frontend y verificar si "params.is\_backoffice" viene con valor "true".
- Añadir el parámetro data-backoffice="true" a la librería JavaScript principal de FastPay en su integración del frontend. Esta librería es la misma que utilizó para realizar la integración del módulo de pago.
- En el ejemplo HTML, solo deberá añadir la parte subrayada en el primer script:

```
<script data-url="/backend" data-backoffice="true" data-amount="3000" data-currency="EUR" data-placeholder="#app" src="https://sandbox.Sipay.es/pwall_app/js/app.js"> </script>
```

## 6. COMPATIBILIDAD CON OTROS NAVEGADORES

Con el fin de aumentar la compatibilidad con más dispositivos, hemos incorporado dos cambios en la integración de la solución con el frontend. En el ejemplo HTML ya están reflejados estos cambios.

El cambio implica quitar todas las llamadas de "addEventListener" y "dispatchEvent".

- Se ha añadido un método "window.PaymentWall.listenTo" que es equivalente a "addEventListener".
- Se ha añadido un método "window.PaymentWall.start()" que reemplaza la llamada "document.dispatchEvent('payment\_wall\_load\_app')".
- Una vez se ha ejecutado el evento "payment\_wall\_loaded" tendremos disponible "window.pwall.dispatch(evento, datos)" equivalente a "document.getElementById('app').dispatchEvent(new CustomEvent(evento, {"detail": datos}))".

### Ejemplo

```
var placeholder=document.getElementById('app');
console.log(placeholder);
window.PaymentWall.listenTo(placeholder, 'payment_wall_load', function(){
  window.PaymentWall.listenTo(placeholder, "payment_wall_loaded", function(){
    window.pwall.dispatch('set_extra_headers', {test_header: "data"});
    window.pwall.dispatch('set_extra_data', {test_data: "data"});
    window.PaymentWall.listenTo(placeholder, 'payment_ok',
      function(ev) {
        document.getElementById('result').style.display = "block";
        document.getElementById('masked_card').innerHTML =
ev.detail.payload.masked_card;
        document.getElementById('order').innerHTML = ev.detail.payload.order;
        document.getElementById('authorizator').innerHTML = ev.detail.payload.authorizator;
        document.getElementById('transaction_id').innerHTML =
ev.detail.payload.transaction_id;
        document.getElementById('currency').innerHTML = ev.detail.payload.currency;
      });
    });
});
window.PaymentWall.start()
```

En la parte de tratamiento del resultado:

Ejemplo
<pre> window.PaymentWall.listenTo(document.getElementById('app'), 'payment_wall_drawn', function(ev) {   window.pwall.dispatch('payment_wall_process_redirect', {"error": {{ error or 'null' }}, "method": "{{ method }}", "request_id": "{{ request_id }}"}) }); </pre>

Nota: Aunque puede mantener la integración actual, recomendamos que realice este cambio lo antes posible. Google Pay no soporta todos los dispositivos. No funcionará en dispositivos con Internet Explorer 11.

## 7. FORMATO DE CAMPOS

Campo	Descripción	Formato	Obligatorio / Opcional
order	Identificación única del pedido. Conocido también como número de ticket. Es el identificador que encontrará en PGC para buscar la operación.	Cadena/string. 6 a 64 caracteres alfanuméricos. <b>Para Bizum el máximo son 12 caracteres.</b> Ejemplo "TICKET0001"	Opcional. En caso de no ser enviado, el sistema generará uno aleatorio.
amount	Importe de la operación. Para las pruebas con Bizum, el importe debe ser superior a 0,50 céntimos.	Cadena/string. Valor entero del importe a cobrar con dos decimales en el caso del euro. Ejemplo: 10 euros serían "1000" porque completamos a 10,00.	Obligatorio.
nonce	Marca de tiempo para asegurar la transacción. Se recomienda que sea un Unix Timestamp.	Cadena/string. 1 a 15 caracteres numéricos. Ejemplo: "157866352904721"	Obligatorio.
group_id	Identificador del usuario donde agruparemos todos sus tokens. Si no queremos agrupar los tokens, utilizaremos el valor 0.  <u>Debe tener cuidado con la gestión de este valor porque si utiliza siempre el mismo valor (distinto de 0) podrían visualizarse los tokens de otros clientes. En ningún caso los datos de la tarjeta estarán completos.</u>	Cadena/string. 1 a 128 caracteres alfanuméricos. El 0 está reservado.	Opcional.

Campo	Descripción	Formato	Obligatorio / Opcional
token	Indicaremos un valor en caso de que queramos almacenar la tarjeta. Este valor puede ser generado por el comercio o puede recibirlo de la respuesta del API generado por SIPAY.	Cadena/string. 1 a 128 caracteres alfanuméricos. Se recomienda que sea un valor impredecible. Ejemplo "TOKEN1".	Opcional.
currency	ISO4217 de la moneda de la transacción.	Cadena/string. Tres caracteres en mayúsculas. Ejemplo: "EUR"	Obligatorio.
notify.result	Es la dirección web del comercio donde se retornará el control cuando se haya realizado un pago que requiera salir de la web del comercio.	Dirección URL con protocolo https. Ejemplo: " <a href="https://www.example.com/">https://www.example.com/</a> ". Es importante que el dominio esté compuesto de tres partes separadas por punto.	Obligatorio.
emv3ds	Datos adicionales opcionales para emv3ds. (PSD2) La inclusión de estos datos reduce en gran medida, en caso de pedir exenciones, la denegación de las mismas y por tanto la petición de challenge.	Objeto de datos EMV3ds. Ver sección "PSD2"	Opcional
sca_exemptions	Exenciones PSD2	En caso de querer aplicar una exención, se deberá especificar "sca_exemptions": "LWV", "TRA" o "COR".	Opcional
reason	Razon de la petición de exención psd2	En caso de especificar exención "MIT", se deberá especificar "sca_exemptions": "MIT" y "reason": "R"	



Campo	Descripción	Formato	Obligatorio / Opcional
phone_number	Número de teléfono con formato +34XXXXXXXXX. Actualmente utilizado para Bizum request-to-pay y el pago en dos pasos.	Con prefijo internacional y carácter +.	Opcional.

Expresión regular para validar la notify.result:

```
(?i)((?https?://|www\d{0,3}[.]?)?[a-z0-9.\-
]+[.](?:(?international)|(?construction)|(?contractors)|(?enterprises)|(?photography)|(?immobilen)
|(?management)|(?technology)|(?directory)|(?education)|(?equipment)|(?institute)|(?marketing)
|(?solutions)|(?builders)|(?clothing)|(?computer)|(?democrat)|(?diamonds)|(?graphics)|(?holding
s)|(?lighting)|(?plumbing)|(?training)|(?ventures)|(?academy)|(?careers)|(?company)|(?domains)
|(?florist)|(?gallery)|(?guitars)|(?holiday)|(?kitchen)|(?recipes)|(?shiksha)|(?singles)|(?support)|(?s
ystems)|(?agency)|(?berlin)|(?camera)|(?center)|(?coffee)|(?estate)|(?kaufen)|(?luxury)|(?monas
h)|(?museum)|(?photos)|(?repair)|(?social)|(?tattoo)|(?travel)|(?viajes)|(?voyage)|(?build)|(?chea
p)|(?codes)|(?dance)|(?email)|(?glass)|(?house)|(?ninja)|(?photo)|(?shoes)|(?solar)|(?today)|(?aer
o)|(?arpa)|(?asia)|(?bike)|(?buzz)|(?camp)|(?club)|(?coop)|(?farm)|(?gift)|(?guru)|(?info)|(?jobs)((
?kiwi)|(?land)|(?limo)|(?link)|(?menu)|(?mobi)|(?moda)|(?name)|(?pics)|(?pink)|(?post)|(?rich)|(?
:ruhr)|(?sexy)|(?tips)|(?wang)|(?wien)|(?zone)|(?biz)|(?cab)|(?cat)|(?ceo)|(?com)|(?edu)|(?gov)|(?i
nt)|(?mil)|(?net)|(?onl)|(?org)|(?pro)|(?red)|(?tel)|(?uno)|(?xxx)|(?ac)|(?ad)|(?ae)|(?af)|(?ag)|(?ai)(
?al)|(?am)|(?an)|(?ao)|(?aq)|(?ar)|(?as)|(?at)|(?au)|(?aw)|(?ax)|(?az)|(?ba)|(?bb)|(?bd)|(?be)|(?bf)
|(?bg)|(?bh)|(?bi)|(?bj)|(?bm)|(?bn)|(?bo)|(?br)|(?bs)|(?bt)|(?bv)|(?bw)|(?by)|(?bz)|(?ca)|(?cc)|(?c
d)|(?cf)|(?cg)|(?ch)|(?ci)|(?ck)|(?cl)|(?cm)|(?cn)|(?co)|(?cr)|(?cu)|(?cv)|(?cw)|(?cx)|(?cy)|(?cz)|(?d
e)|(?dj)|(?dk)|(?dm)|(?do)|(?dz)|(?ec)|(?ee)|(?eg)|(?er)|(?es)|(?et)|(?eu)|(?fi)|(?fj)|(?fk)|(?fm)|(?fo)
|(?fr)|(?ga)|(?gb)|(?gd)|(?ge)|(?gf)|(?gg)|(?gh)|(?gi)|(?gl)|(?gm)|(?gn)|(?gp)|(?gq)|(?gr)|(?gs)|(?g
t)|(?gu)|(?gw)|(?gy)|(?hk)|(?hm)|(?hn)|(?hr)|(?ht)|(?hu)|(?id)|(?ie)|(?il)|(?im)|(?in)|(?io)|(?iq)|(?ir)
|(?is)|(?it)|(?je)|(?jm)|(?jo)|(?jp)|(?ke)|(?kg)|(?kh)|(?ki)|(?km)|(?kn)|(?kp)|(?kr)|(?kw)|(?ky)|(?kz)((
?la)|(?lb)|(?lc)|(?li)|(?lk)|(?lr)|(?ls)|(?lt)|(?lu)|(?lv)|(?ly)|(?ma)|(?mc)|(?md)|(?me)|(?mg)|(?mh)|(?
mk)|(?ml)|(?mm)|(?mn)|(?mo)|(?mp)|(?mq)|(?mr)|(?ms)|(?mt)|(?mu)|(?mv)|(?mw)|(?mx)|(?my)
|(?mz)|(?na)|(?nc)|(?ne)|(?nf)|(?ng)|(?ni)|(?nl)|(?no)|(?np)|(?nr)|(?nu)|(?nz)|(?om)|(?pa)|(?pe)|(?
pf)|(?pg)|(?ph)|(?pk)|(?pl)|(?pm)|(?pn)|(?pr)|(?ps)|(?pt)|(?pw)|(?py)|(?qa)|(?re)|(?ro)|(?rs)|(?ru)((
?rw)|(?sa)|(?sb)|(?sc)|(?sd)|(?se)|(?sg)|(?sh)|(?si)|(?sj)|(?sk)|(?sl)|(?sm)|(?sn)|(?so)|(?sr)|(?st)|(?su)
)|(?sv)|(?sx)|(?sy)|(?sz)|(?tc)|(?td)|(?tf)|(?tg)|(?th)|(?tj)|(?tk)|(?tl)|(?tm)|(?tn)|(?to)|(?tp)|(?tr)|(?tt)
(?tv)|(?tw)|(?tz)|(?ua)|(?ug)|(?uk)|(?us)|(?uy)|(?uz)|(?va)|(?vc)|(?ve)|(?vg)|(?vi)|(?vn)|(?vu)|(?wf)((
?ws)|(?ye)|(?yt)|(?za)|(?zm)|(?zw))(?/[^\s()<>+[\s`!(){};:\\".,<?\\xab\\xbb\\u201c\\u201d\\u2018\\u201
9])?)
```

## 8. PSD2

El nuevo parámetro “emv3ds” permite enviar datos complementarios para la decisión por parte del emisor sobre la aplicación de la exención pedida. De estos datos es recomendable rellenar cuantos más de ellos disponibles.

Atributo JSON (string)	Longitud String	Descripción
account_additional_info		Información adicional sobre el usuario
account_info		Información de la cuenta
account_age_indicator	2/A-N	Periodo de tiempo que el titular de la tarjeta ha tenido la cuenta en el comercio: 01=Sin cuenta (guest check-out); 02=Creada durante la transacción; 03=Menos de 30 días; 04=30-60 días; 05=Más de 60 días
account_modification_date	8/N	Fecha en que se modificó por última vez la cuenta del titular de la tarjeta en el comercio, incluida la dirección de facturación o de envío, la nueva cuenta de pago o los nuevos usuarios agregados. Formato: YYYYMMDD
account_modification_indicator	2/A-N	Período de tiempo transcurrido desde que se modificó por última vez la información de la cuenta del titular de la tarjeta en el comercio, incluida la dirección de facturación o de envío, la nueva cuenta de pago o los nuevos usuarios agregados: 01=Modificado en esta transacción; 02=Menos de 30 días; 03=30-60 días; 04=Más de 60 días
account_creation_date	9/N	Fecha en que el titular de la tarjeta abrió la cuenta en el comercio. Formato: YYYYMMDD
account_pw_change_date	8/N	Fecha en que el titular tuvo un cambio de contraseña en la cuenta del comercio o un restablecimiento de la cuenta. Formato: YYYYMMDD
account_pw_change_indicator	2/A-N	Indicador de cambio de contraseña: 01=Sin cambios; 02=Cambió durante esta operación; 03=Menos de 30 días; 04=30-60 días; 05=Más de 60 días
account_purchase_number	4/N	Número de compras con esta cuenta durante los últimos seis meses
transactions_day	3/N	Número de transacciones (exitosas y abandonadas) para esta cuenta del titular de la tarjeta en el comercio en todas las cuentas de pago en las últimas 24 horas

Atributo JSON (string)	Longitud String	Descripción
transactions_year	3/N	Número de transacciones (exitosas y abandonadas) para esta cuenta del titular de la tarjeta en el comercio en todas las cuentas de pago del año anterior
account_age_date	8/N	Fecha en que la cuenta de pago se inscribió en la cuenta del titular de la tarjeta en el comercio. Formato: YYYYMMDD
pay_account_creation_date	8/N	Fecha en que la cuenta de pago se inscribió en la cuenta del titular de la tarjeta en el comercio. Formato: YYYYMMDD
pay_account_indicator	2/A-N	Indica el período de tiempo que la cuenta de pago se inscribió en la cuenta del titular de la tarjeta en el comercio: 01=Sin cuenta (guest check-out); 02=Durante la transacción; 03=Menos de 30 días; 04=30-60 días; 05=Más de 60 días
address_first_use_date	8/N	Fecha en que la dirección de envío utilizada para esta transacción se utilizó por primera vez con el comercio. Formato: YYYYMMDD
address_first_use_indicator	2/A-N	Indica cuándo la dirección de envío utilizada para esta transacción se utilizó por primera vez con el comercio: 01=Esta transacción; 02=Menos de 30 días; 03=30-60 días; 04=Más de 60 días
shipment_name_indicator	2/A-N	Indica si el Nombre del titular de la tarjeta en la cuenta es idéntico al Nombre de envío utilizado para esta transacción: 01=Nombre de la cuenta idéntico al nombre del envío; 02=Nombre de la cuenta diferente al nombre de envío
suspicious_activity_indicator	2/A-N	Indica si el comercio ha experimentado actividad sospechosa (incluido fraude anterior) en la cuenta del titular de la tarjeta: 01=No se han observado actividades sospechosas; 02=Se han observado actividades sospechosas
addresses_match	1/A-N	Indicador de coincidencia de dirección
billing_city	50/A-N	Ciudad de la dirección de facturación
billing_country	3/N	País de la dirección de facturación
billing_address_1	50/A-N	Dirección de facturación (línea 1)
billing_address_2	50/A-N	Dirección de facturación (línea 2)
billing_address_3	50/A-N	Dirección de facturación (línea 3)

Atributo JSON (string)	Longitud String	Descripción
billing_postcode	16/A-N	Código postal de la dirección de facturación
cardholder_mobile_phone		Teléfono móvil del dueño de la tarjeta
prefix	1-3/N	Indicativo del país correspondiente al número de teléfono, según
number	15/N	Numeración correspondiente al número de teléfono
cardholder_home_phone*		Teléfono fijo del dueño de la tarjeta
prefix	1-3/N	Indicativo del país correspondiente al número de teléfono, según
number	15/N	Numeración correspondiente al número de teléfono
cardholder_work_phone		Teléfono corporativo del dueño de la tarjeta
prefix	1-3/N	Indicativo del país correspondiente al número de teléfono, según
number	15/N	Numeración correspondiente al número de teléfono
merchant_risk_indicator	JSON object	Indicador de riesgo del comerciante
delivery_email_address	254/A-N	Para la entrega electrónica, la dirección de correo electrónico a la que se entregó la mercancía
delivery_timeframe	2/A-N	Indica el plazo de entrega de la mercancía: 01=Envío electrónico; 02=Entrega en el mismo día; 03=Entrega de la noche a la mañana (overnight shipping); 04=Dos o más días para la entrega
amount_in_giftcards	15/A-N	Para compras de tarjetas prepago o tarjetas regalo, el importe total de la compra en unidades principales (por ejemplo, USD 123.45 es 123)
amount_of_giftcards	2/N	Para a compras de tarjeta prepago o tarjetas regalo, recuento total de tarjetas prepago o tarjetas/códigos regalo comprados
preorder_date	8/N	Para una compra preordenada, la fecha prevista de disponibilidad de la mercancía. Formato: YYYYMMDD
preorder_availability_indicator	2/A-N	Indica si el titular de la tarjeta realiza un pedido con disponibilidad o fecha de lanzamiento futuros. Valores aceptados: 01=Pedido disponible; 02=Disponible próximamente

Atributo JSON (string)	Longitud String	Descripción
reorder_indicator	2/A-N	Indica si el titular de la tarjeta está reordenando mercancía comprada previamente: 01=Orden creada por primera vez; 02=Reordenado
shipping_indicator	2/A-N	Indica el método de envío elegido para la transacción. Los comercios deben elegir el código del indicador de envío que describa con mayor precisión la transacción específica del titular de la tarjeta, no su actividad comercial en general. Si se incluyen uno o más artículos en la venta, utilice el código del Indicador de envío para los bienes físicos, o si todos los productos son digitales, utilice el código del Indicador de envío que describe el artículo más caro: 01=Envío a la dirección del dueño de la tarjeta; 02=Envío a otra dirección verificada en los archivos del comerciante; 03=Envío a una dirección diferente a la del dueño de la tarjeta; 04="Ship to Store" / Recogida en el local (la dirección de la tienda se rellenará en los campos de dirección de envío); 05=Bienes digitales (incluye servicios en línea, tarjetas electrónicas de regalo y códigos canjeables); 06=Billetes de viaje y de eventos, no enviados; 07=Otros (por ejemplo, videojuegos, servicios digitales no enviados, suscripciones a emedia, etc.)
shipping_city	50/A-N	Ciudad de la dirección de envío
shipping_country	3/N	País de la dirección de envío
shipping_address_1	50/A-N	Dirección de envío (línea 1)
shipping_address_2	50/A-N	Dirección de envío (línea 2)
shipping_address_3	50/A-N	Dirección de envío (línea 3)
shipping_postcode	16/A-N	Código postal de la dirección de envío

## Ejemplo

```
"emv3ds": {  
  
  "account_additional_info": "more info",  
  "account_info": {  
    "account_age_indicator": "02",  
    "account_modification_date": "03032022",  
    "account_modification_indicator": "02",  
    "account_creation_date": "01012022",  
    "account_pw_change_date": "02022022",  
    "account_pw_change_indicator": "01",  
    "account_purchase_number": "20",  
    "transactions_day": "1",  
    "transactions_year": "4",  
    "account_age_date": "01022022",  
    "pay_account_creation_date": "03012022",  
    "pay_account_indicator": "01",  
    "address_first_use_date": "01012022",  
    "address_first_use_indicator": "01",  
    "shipment_name_indicator": "01",  
    "suspicious_activity_indicator": "02"  
  },  
  "addresses_match": "N",  
  "billing_city": "Mississippi",  
  "billing_country": "020",  
  "billing_address_1": "calle cualquiera",  
  "billing_address_2": "piso cualquiera",  
  "billing_address_3": "puerta cualquiera",  
  "billing_postcode": "28080",  
  "cardholder_mobile_phone": {  
    "prefix": "1",  
    "number": "666778899"  
  },  
  "cardholder_home_phone": {  
    "prefix": "11",  
    "number": "666778899"  
  },  
  "cardholder_work_phone": {  
    "prefix": "111",  
    "number": "666778899"  
  },  
  "merchant_risk_indicator": {  
    "delivery_email_address": "paco@paco.com",  
    "delivery_timeframe": "02",  
    "amount_in_giftcards": "10",  
    "amount_of_giftcards": "01",  
    "preorder_date": "20211220",  
    "preorder_availability_indicator": "01",  
    "reorder_indicator": "02",  
    "shipping_indicator": "01"  
  },  
  "shipping_city": "trololo",  
  "shipping_country": "100",  
  "shipping_address_1": "mi calle 4",  
  "shipping_address_2": "mi casa 3",  
  "shipping_address_3": "mi piso 2",  
  "shipping_postcode": "02"  
}  
}
```

## 9. CONSIDERACIONES PARA LA ACTIVACIÓN EN PRODUCCIÓN (LIVE)

Debe considerar que la integración técnica que haya realizado debería permitir configurar los siguientes valores:

- Base URL de las llamadas API: la base de todas sus llamadas API en sandbox están utilizando este dominio: [sandbox.sipay.es](https://sandbox.sipay.es). Cuando cambie de entorno a LIVE deberá utilizar [live.sipay.es](https://live.sipay.es)
- Key: se le proporcionará un valor diferente por cada entorno.
- Resource: se le proporcionará un valor diferente por cada entorno.
- Secret: se le proporcionará un valor diferente por cada entorno.



**© Sipayplus. Todos los derechos reservados.**

Este documento y, en su caso, cualquier fichero anexo al mismo, contiene información privada y estrictamente confidencial exclusivamente dirigida al destinatario(s). Dicha información es propiedad de SIPAY PLUS, S.L. y/o cualquier empresa participada o relacionada con las mismas, en adelante la empresa. Se prohíbe realizar copias del presente documento, así como cualquier divulgación, o acción o medida con respecto a dicha información a todas aquellas personas o entidades a quienes no vaya dirigida, tanto parcial como totalmente sin el permiso por escrito de la empresa. Si Usted ha recibido esta información por error, se ruega lo notifique inmediatamente al emisor y lo elimine de todos los ordenadores.

Todas las afirmaciones realizadas en el presente documento pueden estar sujetas a modificaciones posteriores en cualquier momento. La empresa no garantiza, representa o asume que el presente documento esté completo y ni que se ajuste completamente al propósito del Cliente pues este puede tener requerimientos diferentes, de un ámbito con perímetro de servicio mayor al reflejado en el presente documento.

**FASTPAY –**

**Especificaciones de Integración Vía API**

C/ San Rafael 1 Portal 2 2ºC  
Edificio Europa III  
28108 Alcobendas - Madrid  
Tel. +34 91 484 10 28

[www.Sipay.es](http://www.Sipay.es)