

Manual técnico de [↗](#)



v3 [↗](#)

Índice de contenido [↗](#)

- 1. Introducción
 - 1.1 Monedas aceptadas
 - 1.2 Confirmaciones
- 2. API Key y Secret Key
 - 2.1 Obtención de la API Key + Secret Key
 - 2.2 Comunicación de las claves
- 3. Autorización
 - 3.1 Ejemplo práctico en JavaScript
 - 3.2 Ejemplo práctico en PHP para crear un invoice
- 4. Callbacks
 - 4.1 Validación de las callbacks
 - 4.2 Política de re-intentos
- 5. Endpoints
 - 5.1 Salud
 - 5.1.1 Ping
 - 5.1.1.1 Parámetros
 - 5.1.1.2 Respuestas
 - 5.2 Currencies
 - 5.2.1 Listar todas las currencies
 - 5.2.1.1 Parámetros
 - 5.2.1.2 Respuestas
 - 5.2.2 Listar currency
 - 5.2.2.1 Parámetros
 - 5.2.2.2 Respuestas
 - 5.2.3 Rates

- 5.2.2.1 Parámetros
 - 5.2.2.2 Respuestas
- 5.3 Balances
 - 5.3.1 Ver saldos
 - 5.3.1.1 Parámetros
 - 5.3.1.2 Respuestas
- 5.4 Invoices
 - 5.4.1 Crear invoice
 - 5.4.1.1 Parámetros
 - 5.4.1.2 Respuestas
 - 5.4.2 Checkout de un invoice
 - 5.4.2.1 Parámetros de la ruta
 - 5.4.2.2 Parámetros del cuerpo
 - 5.4.2.3 Respuestas
 - 5.4.3 Consultar un invoice
 - 5.4.2.1 Parámetros
 - 5.4.2.3 Respuestas
 - 5.4.4 Callbacks
 - 5.4.4.1 Invoice creado
 - 5.4.4.2 Invoice activado
 - 5.4.4.3 Pago detectado
 - 5.4.4.4 Pago completado
 - 5.4.4.5 Pago expirado
- 6. Pagos on-demand
 - 6.1 Re-utilización de direcciones frente a la generación de nuevas direcciones
 - 6.2 Formularios de pago
 - 6.2.1 ¿Qué es una wallet crypto en formato URI?
 - 6.2.2 ¿Por qué debo presentarle al usuario su dirección como un enlace clicable?
 - 6.2.3 ¿Qué es un código QR?
 - 6.2.3.1 Cómo crear un código QR
 - 6.2.5 Mejores prácticas

1. Introducción

 **Bienvenid@ a Bit2Me Commerce**  

Bit2Me Commerce es uno de los productos más demandados en los últimos tiempos. En esta nueva versión reducimos la complejidad del producto, facilitando a nuestros clientes una API sencilla de usar y fácil de entender, para permitirle cobrar en su negocio con *crypto* pudiendo transformarlas a euros en el momento.

1.1 Monedas aceptadas

Nombre	Tipo	Red
BTC	Crypto	Bitcoin
ETH	Crypto	Ethereum

LTC	Crypto	Litecoin
BCH	Crypto	Bitcoin Cash
XRP	Crypto	Ripple
USDT	Crypto	Ethereum, Binance Smart Chain, Polygon
USDC	Crypto	Ethereum, Binance Smart Chain, Polygon
TRX	Crypto	Tron
EUR	Fiat	-
USD	Fiat	-

1.2 Confirmaciones

Moneda	Red	Confirmaciones
BTC	Bitcoin	Hasta 200 EUR: 1 confirmación 200-1000 EUR: 2 confirmaciones 1000-5000 EUR: 3 confirmaciones Mayor a 5000 EUR: 6 confirmaciones
BTC	Lightning Network	Instantáneo
ETH, USDC, USDT	Ethereum	Hasta 100 EUR: 1 confirmación 100-500 EUR: 2 confirmaciones 500-1000 EUR: 3 confirmaciones 1000-5000 EUR: 5 confirmaciones Mayor a 5000 EUR: 8 confirmaciones
USDC, USDT	Binance Smart Chain	Hasta 100 EUR: 1 confirmación 100-500 EUR: 2 confirmaciones 500-1000 EUR: 3 confirmaciones 1000-5000 EUR: 5 confirmaciones Mayor a 5000 EUR: 8 confirmaciones
USDC, USDT	Polygon	Hasta 100 EUR: 3 confirmaciones 100-500 EUR: 5 confirmaciones 500-1000 EUR: 7 confirmaciones 1000-5000 EUR: 10 confirmaciones Mayor a 5000 EUR: 20 confirmaciones
BCH	Bitcoin Cash	Hasta 200 EUR: 2 confirmaciones 200- 1000 EUR: 3 confirmaciones 1000- 5000 EUR: 4 confirmaciones
LTC	Litecoin	Hasta 200 EUR: 1 confirmación 200- 1000 EUR: 2 confirmaciones 1000- 5000 EUR: 3 confirmaciones Mayor a 5000 EUR: 6 confirmaciones
XRP	Ripple	1 confirmación

TRX	Tron	Hasta 100 EUR: 5 confirmaciones 100-1000 EUR: 9 confirmaciones Mayor a 1000 EUR: 20 confirmaciones
-----	------	--

2. API Key y Secret Key [↗](#)

2.1 Obtención de la API Key + Secret Key [↗](#)

Después de firmar el contrato de Bit2Me Commerce y registrarse como empresa en Bit2Me, se le dará de alta como Merchant oficial y se le enviarán 2 claves:

- *Api Key*: Se trata de una clave de 512 caracteres para ser autenticado en el servicio de Bit2Me Commerce.
- *Secret Key*: Esta clave **NUNCA** debe compartirla, pues es la llave de firma que le permitirá interactuar con Bit2Me Commerce tal y como se indica en el punto 3.

2.2 Comunicación de las claves [↗](#)

Bit2Me le enviará un correo confidencial y adjunto a este, un zip protegido por una contraseña, la cual se le comunicará usando un canal externo (Telegram, Whatsapp, SMS). Finalmente, deberá utilizar la contraseña recibida mediante el canal externo para descifrar el activo.

3. Autorización [↗](#)

Para realizar cualquier petición a la *API REST* de Bit2Me Commerce, se necesita codificar de la siguiente manera:

- Se debe añadir el parámetro `b2m-processing-key` en las cabeceras de la petición, cuyo valor será su *API Key*.
- También se debe añadir el parámetro `b2m-secret-key` en las cabeceras de la petición, cuyo valor será el valor codificado en `sha256` resultante de combinar la `Secret Key` y el `payload`, es decir, la información que se quiere enviar.
- Por último, se debe añadir una última cabecera, indicando que el *Content-Type* es tipo JSON de la siguiente manera: `"Content-Type": "application/json"`.

3.1 Ejemplo práctico en JavaScript [↗](#)

```

1  const { createHmac } = require('crypto');
2  const axios = require('axios');
3
4  const signSecret = (payload, secret) => createHmac('sha256', secret).update(JSON.stringify(payload)).digest('hex');
5
6  const key = "*****";
7  const secret = "*****";
8
9  const payload = {};
10
11 const baseUrl = 'https://gateway.bit2me.com/v3/commerce';
12 const endpoint = '/merchants/id/balances';
13
14 const options = {
15   method: 'GET', // Bit2Me Commerce has GET and POST endpoints
16   url: baseUrl + endpoint,
17   headers: {
18     'content-type': 'application/json',
19     'b2m-processing-key': key,
20     'b2m-secret-key': signSecret(payload, secret),
21   }
22 };
23
24 axios
```

```

25 .request(options)
26 .then(function (response) {
27     console.log(response.data);
28 })
29 .catch(function (error) {
30     console.error(error);
31 });

```

3.2 Ejemplo práctico en PHP para crear un invoice

```

1  private $key = "*****";
2  private $secret = "*****";
3
4  const API_URL = 'https://gateway.bit2me.com/';
5  const API_VERSION = 'v3/commerce';
6
7  try {
8      $EndPoint = '/merchants/id/balances';
9
10     $postData = '{}';
11
12     $signature = hash_hmac('sha256', $postData, $this->secret);
13
14     $url = self::API_URL . self::API_VERSION . '/' . $EndPoint;
15     $agent = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.13
16
17     $ch = curl_init($url);
18     curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
19     curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 1);
20     curl_setopt($ch, CURLOPT_VERBOSE, 0);
21     curl_setopt($ch, CURLOPT_USERAGENT, $userAgent);
22     curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
23     curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
24     curl_setopt($ch, CURLOPT_HTTPHEADER, [
25         'b2m-processing-key: ' . $this->key,
26         'b2m-secret-key: ' . $signature,
27         'Content-Type: application/json'
28     ]);
29
30     $result = curl_exec($ch);
31     curl_close($ch);
32
33     $result = json_decode($result);
34     var_dump($result);
35
36 } catch (\Exception $e) {
37     var_dump($e);
38 }
39
40 exit;

```

4. Callbacks

Los pagos asíncronos son muy populares en el mundo *crypto* y nuestro sistema se comunica con usted con las llamadas *callbacks*.

A la hora de registrarse como merchant en Bit2Me Commerce podrá indicar la URI del *callback* a la que quiere recibir las respuestas. Al procesar los pagos, Bit2Me Commerce le enviará notificaciones en formato *JSON* a esta URL con toda la información necesaria sobre la transacción, como su importe y estado.

Nuestras *callbacks* contienen toda la información importante sobre la transacción:

- Estado
- Moneda
- Importe
- Dirección
- Etc...

4.1 Validación de las *callbacks*

Para proporcionar autenticación para las *callbacks*, Bit2Me Commerce le envía en el body un `token` que será el mismo que nos indicó usted a la hora de crear el invoice.

Debe validar el token para evitar el fraude.

Si la *callback* se valida con éxito, su sistema debe responder con el código `200 OK HTTP`. No se necesitan parámetros adicionales en la respuesta del cuerpo.

4.2 Política de re-intentos

Número de re-intentos	Envío cada
Menor o igual a 5	1 minuto
Entre 5 y 10	5 minutos
Entre 10 y 15	10 minutos
Entre 15 y 20	20 minutos
Entre 20 y 25	30 minutos
Entre 25 y 30	1 hora
Entre 30 y 35	5 horas
Entre 35 y 40	1 día

Si el número de re-intentos es mayor a 40, el *callback* se cancelará.

Tras enviar el *callback* Bit2Me Commerce espera una respuesta durante 20 segundos.

5. Endpoints

5.1 Salud

5.1.1 Ping

Tipo	URI	Descripción
GET	https://gateway.bit2me.com/v3/commerce/ping	Comprueba si la <i>API</i> está en funcionamiento.

5.1.1.1 Parámetros [↗](#)

No tiene.

5.1.1.2 Respuestas [↗](#)

✓ 200

```
1 {
2   "status": "ok",
3   "info": {
4     "app": {
5       "status": "up"
6     }
7   },
8   "error": {},
9   "details": {
10    "app": {
11      "status": "up"
12    }
13  }
14 }
```

✗ 50x - Error del servidor

5.2 Currencies [↗](#)

5.2.1 Listar todas las currencies [↗](#)

Tipo	URI	Descripción
GET	https://gateway.bit2me.com/v3/commerce/currencies	Devuelve una lista de todas las monedas admitidas.

5.2.1.1 Parámetros [↗](#)

No tiene

5.2.1.2 Respuestas [↗](#)

✓ 200

```
1 [
2   {
3     currencyId: 'BCH',
4     name: 'Bitcoin Cash',
5     type: 'crypto',
6     receiveType: true,
7     networks: [ [Object] ],
8     imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/bch-circle-solid-default.svg'
9   },
```

```
10 {
11   currencyId: 'BTC',
12   name: 'Bitcoin',
13   type: 'crypto',
14   receiveType: true,
15   networks: [ [Object], [Object] ],
16   imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/btc-circle-solid-default.svg'
17 },
18 {
19   currencyId: 'ETH',
20   name: 'Ether',
21   type: 'crypto',
22   receiveType: true,
23   networks: [ [Object] ],
24   imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/eth-circle-solid-default.svg'
25 },
26 {
27   currencyId: 'LTC',
28   name: 'Litecoin',
29   type: 'crypto',
30   receiveType: true,
31   networks: [ [Object] ],
32   imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/ltc-circle-solid-default.svg'
33 },
34 {
35   currencyId: 'TRX',
36   name: 'Tron',
37   type: 'crypto',
38   receiveType: false,
39   networks: [ [Object] ],
40   imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/trx-circle-solid-default.svg'
41 },
42 {
43   currencyId: 'USDC',
44   name: 'USD Coin',
45   type: 'crypto',
46   receiveType: false,
47   networks: [ [Object], [Object], [Object] ],
48   imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/usdc-circle-solid-default.svg'
49 },
50 {
51   currencyId: 'USDT',
52   name: 'USD Tether',
53   type: 'crypto',
54   receiveType: true,
55   networks: [ [Object], [Object], [Object] ],
56   imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/usdt-circle-solid-default.svg'
57 },
58 {
59   currencyId: 'XRP',
60   name: 'Ripple',
61   type: 'crypto',
62   receiveType: true,
63   networks: [ [Object] ],
64   imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/xrp-circle-solid-default.svg'
65 },
66 {
67   currencyId: 'EUR',
```

```

68     name: 'Euro',
69     type: 'fiat',
70     receiveType: true,
71     networks: [],
72     imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/eur-circle-solid-default.svg'
73 },
74 {
75     currencyId: 'USD',
76     name: 'United States Dollar',
77     type: 'fiat',
78     receiveType: true,
79     networks: [],
80     imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/usd-circle-solid-default.svg'
81 }
82 ]

```

✖ 50x - Error del servidor

5.2.2 Listar currency [↗](#)

Tipo	URI	Descripción
GET	https://gateway.bit2me.com/v3/currencies/{id}	Devuelve la información de una moneda en concreto.

5.2.2.1 Parámetros [↗](#)

Nombre	Tipo	Descripción
id	string	Código ISO de la moneda de la que se quiere obtener información. Por ejemplo: BTC

5.2.2.2 Respuestas [↗](#)

✔ 200

```

1  {
2    currencyId: 'BTC',
3    name: 'Bitcoin',
4    type: 'crypto',
5    receiveType: true,
6    networks: [
7      { platformId: 'bitcoin', name: 'Bitcoin' },
8      { platformId: 'lightningNetwork', name: 'Lightning Network' }
9    ],
10   imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/btc-circle-solid-default.svg'
11 }

```

✖ 400

```

1 {
2   statusCode: 400,
3   error: 'Bad Request',
4   message: "currency 'UST' is not supported yet",
5   data: [Object],
6   reqId: '83a69807-578a-4992-b869-d27e639fa063'
7 }

```

❌ 50x - Error del servidor

5.2.3 Rates [↗](#)

Tipo	URI	Descripción
GET	https://gateway.bit2me.com/v3/commerce/currencies/{id}/rate/{told}	Devuelve el precio ATM del par de monedas.

5.2.2.1 Parámetros [↗](#)

Nombre	Tipo	Descripción
id	string	Código ISO de la primera moneda. Por ejemplo: BTC
toId	string	Código ISO de la segunda moneda. Por ejemplo: EUR

5.2.2.2 Respuestas [↗](#)

✅ 200

```

1 {
2   currencyFrom: {
3     currencyId: 'BTC',
4     name: 'Bitcoin',
5     type: 'crypto',
6     receiveType: true,
7     networks: [ [Object], [Object] ],
8     imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/btc-circle-solid-default.svg'
9   },
10  currencyTo: {
11    currencyId: 'EUR',
12    name: 'Euro',
13    type: 'fiat',
14    receiveType: true,
15    networks: [],
16    imageUrl: 'https://assets.bit2me.com/assets/images/crypto-icons/v7/eur-circle-solid-default.svg'
17  },
18  rate: 26581.52
19 }

```

✖ 400

```
1 {
2   statusCode: 400,
3   error: 'Bad Request',
4   message: "currency 'PEPE' is not supported yet",
5   data: [Object],
6   reqId: 'f2796042-f5c0-4ba8-90dd-7e40a5b5e25c'
7 }
```

✖ 50x - Error del servidor

5.3 Balances [🔗](#)

5.3.1 Ver saldos [🔗](#)

Este endpoint está protegido por el [sistema de autenticación de Bit2Me Commerce](#).

Tipo	URI	Descripción
GET	https://gateway.bit2me.com/v3/commerce/merchants/id/balances	Devuelve el listado de saldos asociado a un merchant.

5.3.1.1 Parámetros [🔗](#)

No tiene

5.3.1.2 Respuestas [🔗](#)

✔ 200

```
1 [
2   {
3     balanceId: '7a4957e7-68ad-4229-8b33-97a10f0f5d75',
4     balance: '500.548051000000000000',
5     currency: { currencyId: 'USDT', type: 'crypto' }
6   },
7   {
8     balanceId: 'bc5ca786-9d89-4919-8f9a-bb692fa6d2ed',
9     balance: '20.590000000000000000',
10    currency: { currencyId: 'EUR', type: 'fiat' }
11  },
12  { balance: 0, currency: { currencyId: 'BCH', type: 'crypto' } },
13  { balance: 0, currency: { currencyId: 'BTC', type: 'crypto' } },
14  { balance: 0, currency: { currencyId: 'ETH', type: 'crypto' } },
15  { balance: 0, currency: { currencyId: 'LTC', type: 'crypto' } },
16  { balance: 0, currency: { currencyId: 'TRX', type: 'crypto' } },
17  { balance: 0, currency: { currencyId: 'USDC', type: 'crypto' } },
18  { balance: 0, currency: { currencyId: 'XRP', type: 'crypto' } },
19  { balance: 0, currency: { currencyId: 'USD', type: 'fiat' } }
20 ]
```

✖ 401 - Unauthorized

```
1 {
2   statusCode: 401,
3   error: 'Unauthorized',
4   message: 'Signature failed',
5   data: [Object],
6   reqId: 'f95bec96-5aa5-449f-96c5-5ebb7b4da1eb'
7 }
```

✖ 50x - Error del servidor

5.4 Invoices [↗](#)

5.4.1 Crear invoice [↗](#)

Este endpoint está protegido por el [sistema de autenticación de Bit2Me Commerce](#).

Tipo	URI	Descripción
POST	https://gateway.bit2me.com/v3/commerce/invoices/	Crea una factura para su cliente por una cantidad específica

5.4.1.1 Parámetros [↗](#)

NOTA: Todos los atributos en los cuales aparezca un asterisco (*) son obligatorios.

Nombre	Tipo	Descripción
title*	string	El título de la factura que se le mostrará a su cliente.
description*	string	La descripción de la factura que se le mostrará a su cliente.
priceCurrency*	string	Código ISO de la moneda en la que el cliente paga la factura. Por ejemplo: EUR .
priceAmount*	string	La cantidad que quiere recibir de su cliente, por ejemplo: 10 . Junto a <code>priceCurrency</code> forman el invoice: 10 EUR .
receiveCurrency	string	Código ISO de la moneda de la factura que quiere recibir. Si no se especifica, toma por defecto <code>priceCurrency</code> realizando así un cambio <i>on-the-fly</i> .
foreignId*	string	Una ID única en su sistema, por ejemplo: BWpSk4yzcuO3nh11 . Recomendado usar un método que genere un ID único en cada invoice.

successUrl*	string	La URL a la que redirigiremos a su cliente en caso de pago de factura exitosa, por ejemplo: https://nombre.empresa.com/url_success
cancelUrl*	string	La URL a la que redirigiremos a su cliente en caso de pago de factura fallida, por ejemplo: https://nombre.empresa.com/url_failed
securityToken*	string	Cadena única con el cual se verifica el callback procedente de nuestro sistema. Recomendado generar un securityToken por cada invoice y almacenarlo para posteriores usos.
purchaserEmail*	string	Si el importe del pago no coincide con el que figura en la factura, enviaremos un correo electrónico a la dirección especificada con instrucciones para la recuperación de fondos, mientras que en el caso de que el pago sea insuficiente, se reembolsará la totalidad del importe. Por último, en el caso de un pago con un importe superior, el destinatario del correo podrá recuperar la diferencia siguiendo las instrucciones. Si el pago es correcto, se envía un correo de pago completado.

5.4.1.2 Respuestas [↗](#)

✓ 200

```

1 {
2   "invoiceId": "666c4e53-9121-40aa-850a-0c9990010d7d",
3   "createdAt": "2023-08-10T07:00:30.080Z",
4   "updatedAt": "2023-08-10T07:00:30.080Z",
5   "foreignId": "BwPsk4yzcu03nh11",
6   "priceAmount": 8,
7   "priceCurrency": "EUR",
8   "status": "new",
9   "customerEmail": "customer@test.com",
10  "receiveCurrencyName": "EUR",
11  "title": "iPhone 12",
12  "description": ".",
13  "successUrl": "https://dominio.com/success",
14  "cancelUrl": "https://dominio.com/cancel",
15  "merchant": {
16    "merchantId": "3e95e3d5-4d0b-47b6-bce2-f40427a5ac7e",
17    "name": "Empresa 1"
18  },
19  "url": "https://commerce.qa.bit2me.com/payment?invoice=666c4e53-9121-40aa-850a-0c9990010d7d"
20 }
```

✖ 401 - Unauthorized

```
1 {
2   statusCode: 401,
3   error: 'Unauthorized',
4   message: 'Signature failed',
5   data: [Object],
6   reqId: 'f95bec96-5aa5-449f-96c5-5ebb7b4da1eb'
7 }
```

✖ 50x - Error del servidor

5.4.2 Checkout de un invoice [↗](#)

Tipo	URI	Descripción
PUT	https://gateway.bit2me.com/v3/commerce/invoices/{id}/checkout	Activa una factura para su pago. El tiempo límite son 20 minutos.

5.4.2.1 Parámetros de la ruta [↗](#)

Nombre	Tipo	Descripción
id	string	uuid de la factura que se quiere activar

5.4.2.2 Parámetros del cuerpo [↗](#)

Nombre	Tipo	Descripción
currencyId	string	Código ISO de la moneda con la que se va a pagar. Por ejemplo: USDT.
networkId	string	ID de la red en la que se quiere pagar. Por ejemplo: polygon.

NOTA: Para obtener tanto la `currencyId` como la `networkId` se recomienda consultar primero con el endpoint `/currencies/{id}`.

5.4.2.3 Respuestas [↗](#)

✔ 200

```
1 {
2   "invoiceId": "666c4e53-9121-40aa-850a-0c9990010d7d",
3   "createdAt": "2023-08-10T07:00:30.080Z",
4   "updatedAt": "2023-08-10T07:05:50.013Z",
5   "expiredAt": "2023-08-10T07:25:50.000Z",
6   "paidAt": null,
7   "foreignId": "BwPsk4yzcu03nh11",
8 }
```

```

8  "priceAmount": "8",
9  "priceCurrency": "EUR",
10 "status": "pending",
11 "customerEmail": "customer@test.com",
12 "receiveCurrencyName": "EUR",
13 "title": "iPhone 12",
14 "description": ".",
15 "successUrl": "https://dominio.com/success",
16 "cancelUrl": "https://dominio.com/cancel",
17 "underpaidAmount": "8.8249",
18 "overpaidAmount": "0",
19 "paymentAddress": "0xb27718181446445cec93f889d6fA8F408825eC07",
20 "paymentRequestUri": "ethereum:0xc2132d05d31c914a87c6611c10748aeb04b58e8f@137/transfer?address=0xb277181814464
21 "payAmount": 8.8249,
22 "payCurrency": "USDT",
23 "merchant": {
24   "merchantId": "3e95e3d5-4d0b-47b6-bce2-f40427a5ac7e",
25   "name": "Empresa 1"
26 },
27 "url": "https://commerce.qa.bit2me.com/payment?invoice=666c4e53-9121-40aa-850a-0c9990010d7d"
28 }

```

✖ 400 - Factura inválida

```

1  {
2    statusCode: 400,
3    error: "Bad Request",
4    message: "'id: 668bd9a4-a404-4195-95f4-3d0a4514b664' is not available",
5    data: [Object],
6    reqId: "e6d94e8b-1038-4c25-b962-820dc34f5287"
7  }

```

✖ 400 - Moneda no soportada

```

1  {
2    statusCode: 400,
3    error: "Bad Request",
4    message: "currency 'ELON' is not supported yet",
5    data: [Object],
6    reqId: "e8ed03aa-515b-4679-8a0b-2b0d638e1e3a"
7  }

```

✖ 400 - Red no soportada

```

1  {
2    statusCode: 400,
3    error: "Bad Request",
4    message: "platform id 'harmony' is not supported yet",
5    data: [Object],
6    reqId: "7ec1a249-23af-4756-be0d-c46bcf7dddb1"
7  }

```

❌ 400 - Par no soportado

```
1 {
2   statusCode: 400,
3   error: "Bad Request",
4   message: "platform id 'bitcoin' does not support currency 'USDT'",
5   data: [Object],
6   reqId: "4c18391b-73b3-4e8c-a85b-30d2c6c7c3d3"
7 }
```

❌ 50x - Error del servidor

5.4.3 Consultar un invoice [🔗](#)

Tipo	URI	Descripción
PUT	https://gateway.bit2me.com/v3/commerce/invoices/{id}/checkout	Recupera información asociada a un invoice.

5.4.2.1 Parámetros [🔗](#)

Nombre	Tipo	Descripción
id	string	uuid de la factura que se quiere consultar

5.4.2.3 Respuestas [🔗](#)

✅ 200

```
1 {
2   "invoiceId": "666c4e53-9121-40aa-850a-0c9990010d7d",
3   "createdAt": "2023-08-10T07:00:30.080Z",
4   "updatedAt": "2023-08-10T07:05:50.013Z",
5   "expiredAt": "2023-08-10T07:25:50.000Z",
6   "paidAt": "2023-08-10T07:07:14.000Z",
7   "foreignId": "BwpSk4yzcu03nh11",
8   "priceAmount": "8",
9   "priceCurrency": "EUR",
10  "status": "paid",
11  "customerEmail": "customer@test.com",
12  "receiveCurrencyName": "EUR",
13  "title": "iPhone 12",
14  "description": ".",
15  "successUrl": "https://dominio.com/success",
16  "cancelUrl": "https://dominio.com/cancel",
17  "underpaidAmount": "0.0000000000000000",
18  "overpaidAmount": "0.0000000000000000",
19  "paymentAddress": "0xb27718181446445cec93F889d6fA8F408825eC07",
20  "paymentRequestUri": "ethereum:0xc2132d05d31c914a87c6611c10748aeb04b58e8f@137/transfer?address=0xb277181814464",
21  "payAmount": 8.8249,
22  "payCurrency": "USDT",
23  "merchant": {
```

```

24     "merchantId": "3e95e3d5-4d0b-47b6-bce2-f40427a5ac7e",
25     "name": "Empresa 1"
26 },
27 "url": "https://commerce.qa.bit2me.com/payment?invoice=666c4e53-9121-40aa-850a-0c9990010d7d"
28 }

```

✖ 400 - Factura inválida

```

1 {
2   statusCode: 400,
3   error: "Bad Request",
4   message: "'id: 668bd9a4-a404-4195-95f4-3d0a4514b664' is not available",
5   data: [Object],
6   reqId: "e6d94e8b-1038-4c25-b962-820dc34f5287"
7 }

```

✖ 50x - Error del servidor

5.4.4 Callbacks [↗](#)

5.4.4.1 Invoice creado [↗](#)

```

1 {
2   "id": "666c4e53-9121-40aa-850a-0c9990010d7d",
3   "foreignId": "BwpSk4yzcu03nh11",
4   "cryptoAddress": {},
5   "currencySent": {},
6   "currencyReceived": {
7     "currency": "EUR"
8   },
9   "token": "sRvf0oCpI26MPP8pJqwx",
10  "transactions": [],
11  "fees": [],
12  "error": [],
13  "status": "new"
14 }

```

5.4.4.2 Invoice activado [↗](#)

```

1 {
2   "id": "666c4e53-9121-40aa-850a-0c9990010d7d",
3   "foreignId": "BwpSk4yzcu03nh11",
4   "cryptoAddress": {
5     "currency": "USDT",
6     "address": "0xb27718181446445cec93F889d6fA8F408825eC07"
7   },
8   "currencySent": {
9     "currency": "USDT",
10    "amount": 8.8249,
11    "remainingAmount": "8.8249"
12  },
13  "currencyReceived": {
14    "currency": "EUR"

```

```

15 },
16 "token": "sRvf0oCpI26MPP8pJqWx",
17 "transactions": [],
18 "fees": [],
19 "error": [],
20 "status": "pending"
21 }

```

5.4.4.3 Pago detectado [↗](#)

```

1 {
2   "id": "666c4e53-9121-40aa-850a-0c9990010d7d",
3   "foreignId": "BwpSk4yzcu03nh11",
4   "cryptoAddress": {
5     "currency": "USDT",
6     "address": "0xb27718181446445cec93F889d6fA8F408825eC07"
7   },
8   "currencySent": {
9     "currency": "USDT",
10    "amount": "8.824900000000000000",
11    "remainingAmount": "0"
12  },
13  "currencyReceived": {
14    "currency": "EUR"
15  },
16  "token": "sRvf0oCpI26MPP8pJqWx",
17  "transactions": [],
18  "fees": [],
19  "error": [],
20  "status": "confirming"
21 }

```

5.4.4.4 Pago completado [↗](#)

```

1 {
2   "id": "666c4e53-9121-40aa-850a-0c9990010d7d",
3   "foreignId": "BwpSk4yzcu03nh11",
4   "cryptoAddress": {
5     "currency": "USDT",
6     "address": "0xb27718181446445cec93F889d6fA8F408825eC07"
7   },
8   "currencySent": {
9     "currency": "USDT",
10    "amount": "8.824900000000000000",
11    "remainingAmount": "0"
12  },
13  "currencyReceived": {
14    "currency": "EUR"
15  },
16  "token": "sRvf0oCpI26MPP8pJqWx",
17  "transactions": [],
18  "fees": [],
19  "error": [],
20  "status": "paid"
21 }

```

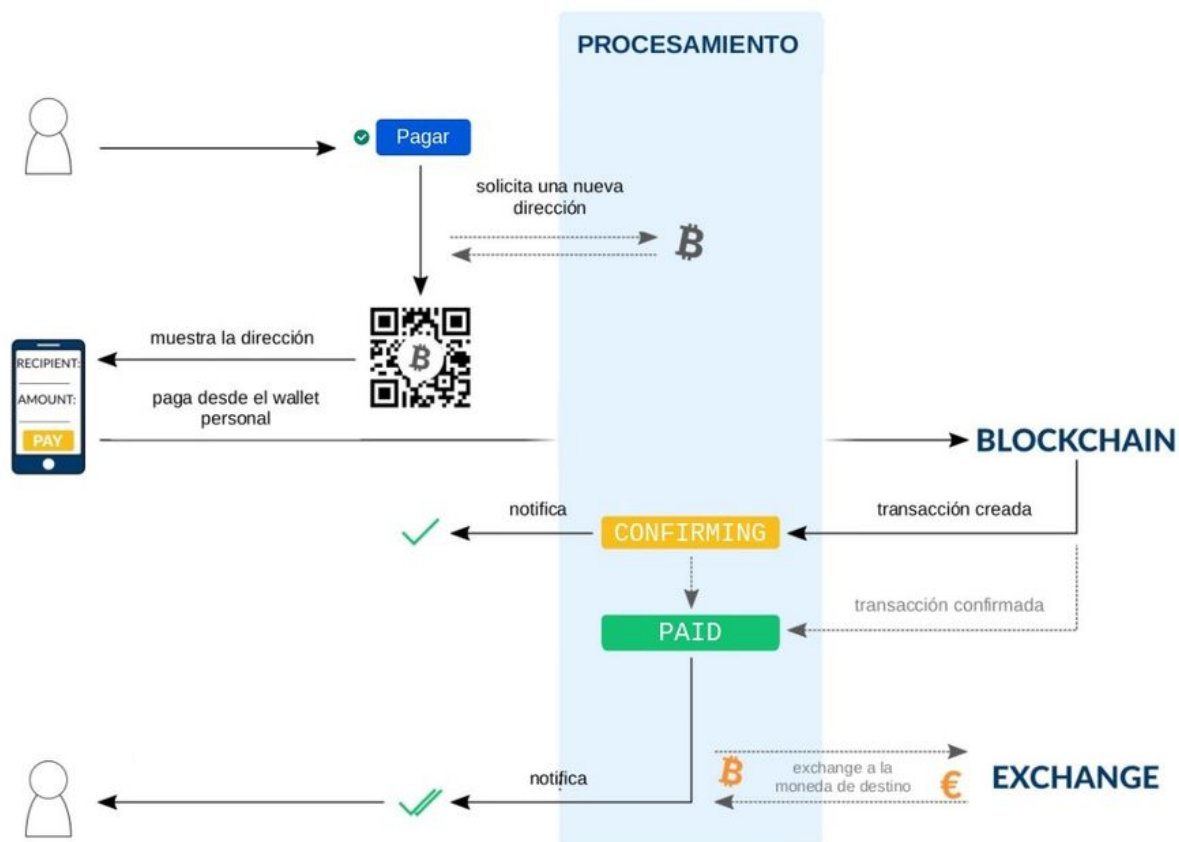
5.4.4.5 Pago expirado [↗](#)

```
1 {
2   "id": "9e8bd9b4-a404-4195-95f4-3d0a4514b333",
3   "foreignId": "FFpSk4yzcu03nh11",
4   "cryptoAddress": {
5     "currency": "USDT",
6     "address": "0xf44718181446445cec93F889d6fA8F408825eD07"
7   },
8   "currencySent": {
9     "currency": "USDT",
10    "amount": "4.443300000000000000",
11    "remainingAmount": "4.4433"
12  },
13  "currencyReceived": {
14    "currency": "EUR"
15  },
16  "token": "sRvf0oCpI26MPP8pJqWx",
17  "transactions": [],
18  "fees": [],
19  "error": [],
20  "status": "expired"
21 }
```

6. Pagos *on-demand* [↗](#)

Cuando se solicita un nuevo invoice de pago, Bit2Me Commerce genera un invoice en estado `new` con el total solicitado en EUR. Después de esto, se redirigirá al usuario a una web donde podrá elegir con que *crypto* pagar, entre todas las que Bit2Me Commerce dispone y entre todas sus redes.

Después de que se acepten los términos de uso y privacidad, se inicia la activación y procesamiento de ese invoice.



Flujo de activación de un invoice

Como se puede observar en el anterior gráfico, Bit2Me Commerce proporciona la infraestructura para que este pago sea posible. Al final del proceso, se le notificará tanto por pantalla como por mail al usuario final.

6.1 Re-utilización de direcciones frente a la generación de nuevas direcciones [↗](#)

Cada invoice tendrá una dirección de depósito distinta, manteniendo la privacidad on-chain del merchant en todo momento. No obstante, Bit2Me Commerce siempre continua controlando los pagos a direcciones de depósito antiguas, incluso después de que se hayan creado otras nuevas.

La razón es que con *crypto*, es imposible borrar o cerrar una dirección existente, por lo que siempre existe la posibilidad de que el usuario haya guardado la dirección anterior en su *wallet* y realice un pago a esta dirección mas adelante.

6.2 Formularios de pago [↗](#)

Si opta por activar usted mismo las facturas, debe saber que en lugar de los métodos de pago clásicos, los usuarios pueden enviar sus fondos mediante *wallets* personales, por lo que para realizar el pago, necesitan un formulario de pago diferente de los formularios de pago clásicos.

El formulario de pago *crypto* es un bloque con información sobre las condiciones de pago y la dirección (*address*) a la que se deben enviar los fondos. La idea principal es proporcionar al usuario la dirección para el depósito incluyendo redes y cantidades.

También sería una buena idea proporcionar códigos QR para que los usuarios puedan utilizarlo de forma automática, abrir su *wallet* y enviar *crypto* con un solo clic.

En esta sección, aprenderá cómo proporcionar al usuario toda la información necesaria, cómo incrustar códigos QR en sus formularios de pagos, y cuáles son las mejores prácticas para hacer que su formulario de pago sea fácil de usar si decide no depender de la solución frontal de Bit2Me Commerce.

6.2.1 ¿Qué es una *wallet* *crypto* en formato URI? [↗](#)

Casi todos los usuarios de PC utilizan enlaces de correo electrónico en los sitios web.

Normalmente, estos enlaces tienen el siguiente formato: `mailto:address@email.com`

Cuando alguien hace clic en el enlace, un cliente del correo se abre de manera automática. El usuario ve un formulario para redactar un nuevo correo que contiene la dirección de correo electrónico del destinatario en el campo «To» (Para). Otros campos también pueden contener datos, como el campo «Field» (Asunto). Se puede utilizar la misma forma de interacción con el usuario con algunas *wallets*.

Por ejemplo, una URI de BTC tiene el siguiente formato: `bitcoin:3AFaCnqriLNxj15kqtp5Pxn8puHfuqbX7W`

Si el usuario tiene una *wallet* en su dispositivo y entra en el enlace, la *wallet* se abrirá con la dirección especificada. Para proceder con el pago, el usuario solo tiene que llevar a cabo una acción: hacer clic en el botón «Pay» (Pagar). La URI de Bitcoin no sólo admite la dirección, sino también otras opciones, como por ejemplo, el importe del pago. Las URIs generadas en otras redes EVM **SÍ** admiten cantidades y tokens.

Ver la norma BIP-00214 para la especificación completa del formato URI de Bitcoin

6.2.2 ¿Por qué debo presentarle al usuario su dirección como un enlace clicable? [↗](#)

Imagine que muestra a su usuario la dirección como una cadena de texto: «Por favor, haga un depósito a través de la siguiente dirección: `3AFaCnqriL-Nxj15kqtp5Pxn8puHfuqbX7W`»

Si el usuario tiene una *wallet* en el mismo dispositivo, necesita:

1. Copiar esta dirección en el portapapeles.
2. Encontrar y abrir su *wallet*.
3. Abrir la sección «Withdraw» (Retirar) o «Send» (Enviar) en la aplicación.
4. Pegar la dirección.
5. Hacer clic en el botón «Send» (Enviar) para transferir sus fondos.

Cuantas más acciones tenga que realizar el usuario, más probable será que no pague. Por eso es preferible mostrar las direcciones como enlaces clicables en su formulario de pago siempre que sea posible. Considere también la posibilidad de utilizar códigos QR para que el usuario pueda escanear rápidamente el enlace con otro dispositivo.

6.2.3 ¿Qué es un código QR? [↗](#)

Un código QR (*quick response code*) es un método visual especial de codificación y transferencia de datos, que suele utilizarse para una interacción semiautomática sencilla y rápida con los dispositivos móviles de los usuarios. Puede contener cualquier dato, incluyendo enlaces. Cuando un usuario escanea el código QR con su teléfono, los datos dentro del código se descodifican. Si es un enlace asociado a una aplicación especial, el teléfono del usuario reaccionará de una manera predefinida. Por ejemplo, puede sugerir al usuario dirigirse a una URI con un navegador web o enviar fondos con su *wallet*. Éste es un ejemplo de código QR:



Las características más importantes del uso de códigos QR son:

- Acceso rápido desde dispositivos móviles
- *Crypto* predefinida
- Un enlace integrado y clicable que abre una *wallet*.

¿Por qué debería utilizar códigos QR en mi formulario de pago?

Imagine que un usuario llega a su formulario de pago a través de su ordenador portátil, y su *wallet* la tiene instalada en su teléfono.

En el peor de los casos, la única manera que tendrá de pagar será escribir de forma manual cada carácter de la dirección. El resultado más probable es que el usuario abandone el formulario sin pagar. Por eso es preferible mostrar un código QR con una dirección codificada.

6.2.3.1 Cómo crear un código QR [↗](#)

Para obtener la mejor experiencia de usuario, basta con convertir una simple URI que contenga una dirección en un código QR.

La manera más sencilla es utilizar el *Google Chart Service*, el cual es un servicio especial que genera códigos QR universales y tiene una *API* muy sencilla para gestionarlo. Según los parámetros de la solicitud, el código QR contendrá los datos requeridos y tendrá un tamaño determinado.

El conjunto mínimo de parámetros es:

- Chs: tamaño del cuadro de imagen (p.ej, 150x150).
- Cht: tipo de cuadro (se requiere **qr**).
- Chl: datos que codificar en su código QR (p.ej, `bitcoin:3AFaCnqriLNxj15kqtp5Pxn8puHfuqbX7W`).
- Choe: el tipo de datos que codificar (La norma **BIP-0021** exige **UTF-8**).

Puede encontrar una descripción más detallada aquí: [QR Codes](#) | [Infographics](#) | [Google for Developers](#)

Este es un ejemplo de un código QR mostrado con la etiqueta `<img>`:

```
1 
```



6.2.5 Mejores prácticas [↗](#)

Para ofrecer la mejor experiencia a sus usuarios, sugerimos combinar el método anterior con el siguiente y usarlos de manera simultánea en su formulario de pago.

El último paso es hacer el código QR clicable

```
1 <a href="bitcoin:3AFaCnqriLNxj15kqtp5Pxn8puHfuqbX7W" target="_blank">
2   
3 </a>
```

A partir de aquí, no importa si su usuario recibe el formulario de pago en su ordenador portátil o en su smartphone. Tienen la oportunidad de hacer pagos fáciles y rápidos en un solo click sin ningún inconveniente.

Para ofrecer la mejor experiencia de usuario, además de un código QR clicable y una dirección clicable, sería buena idea incluir en su formulario de pago la siguiente información:

- Tipo de cambio actual o aproximado a *fiat*
- advertencia sobre la *crypto* que se utilizará con su dirección bien especificada.