



Referencia de Uso: API Banco Plaza

Clientes

Enero 2024

Versión 1.0.0

Tabla de Contenido

Tabla de Contenido	2
Datos de Dominio	3
Procedimiento para el firmado de las peticiones	4
Definición: API-KEY	4
Definición: API-KEY-SECRET	4
Definición: NONCE	4
Definición: API-SIGNATURE	5
Resumen de Operaciones	6
Clientes	6
Recurso: /v0/clientes/{id}	7
Argumentos de Entrada	7
Descripción	7
Estructura de Salida	8
Códigos de Respuesta	9
Recurso: /v0/clientes/{id}	10
Argumentos de Entrada	10
Descripción	10
Estructura de Salida	11
Códigos de Respuesta	12

Datos de Dominio

Todos los recursos descritos en el presente documento son relativos al siguiente dominio:

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	apiqa.bancoplaza.com	8585
Productivo	HTTPS	api.bancoplaza.com	8282

Procedimiento para el firmado de las peticiones

Todas aquellas peticiones a recursos que requieran autenticación (ver la sección [Resumen de Operaciones](#)) deben ser firmados usando un algoritmo HMAC basado en SHA-384. Éste es el mecanismo utilizado para validar que las peticiones provienen de una fuente de confianza.

Para ello, el cliente debe contar dos elementos importantes: Un `api-key` y un `api-key-secret`.

Definición: API-KEY

Llave única de 32 caracteres que identifica al cliente que realiza la petición sobre el recurso (URL). **Debe ser enviado en la cabecera de cada petición que requiera autenticación.** Ejemplo:

```
api-key = f09a0c81d70542deb6e6bfb66d24c845
```

Este parámetro es generado por el cliente a través del portal **Tu Plaza Línea Empresa**. Es de uso exclusivo del cliente.

Definición: API-KEY-SECRET

Llave única de 32 caracteres que hace las veces de un código secreto con el que el cliente firma todas las peticiones que envía al banco. Dado que es elemento de seguridad que se encuentra en custodia tanto del cliente como del banco, sirve entonces para comprobar que toda petición que haya sido firmada con el mismo, provenga del dueño legítimo, es decir, del cliente.

Éste parámetro **no se envía** en ninguna sección de la petición HTTP. Como se indicó, solo se usa para firmar digitalmente la petición.

En términos de estructura, es idéntico al `api-key`. Por ejemplo: `bd18e09732874b04b4046f9c311426e3`.

Definición: NONCE

Cada petición requiere también de un nonce. Un nonce es un número entero que se debe incrementar con cada petición. Su valor, en cada petición, debe ser mayor que el de la petición previa. Se usa para garantizar que la firma de cada petición sea diferente, de tal modo que evita vectores de ataque que eventualmente puedan copiar la estructura de una petición entera y enviarla en un momento posterior. El nonce debe enviarse como un dato en la cabecera de la petición. Por ejemplo:

```
nonce = 616766488366836
```

La forma recomendada para la generación un nonce es a través de una función que retorne un unix timestamp en milisegundos. Por ejemplo:

- En Javascript: `Date.now()`
- En Java: `System.currentTimeMillis()`
- En PHP: `round(microtime(true) * 1000)`

Definición: API-SIGNATURE

Se refiere a la firma de la petición propiamente dicha. Para firmar cada petición se debe:

1. Concatenar el path del recurso al que se le alude en la petición, con el nonce y con el cuerpo (JSON) de la petición.
2. Firmar el resultado de la concatenación anterior con el `api-key-secret` haciendo uso de una función `HMAC.SHA-384`.
3. El resultado de la función `HMAC.SHA-384` (48 bytes) debe ser convertido a una cadena hexadecimal (96 caracteres). Dicha cadena **es lo que se coloca en la cabecera** de la petición bajo el parámetro `api-signature`.

A continuación, se muestra un código de ejemplo en Javascript:

```
1  const CryptoJS = require('crypto-js') // Standard JavaScript cryptography library
2  const request = require('request') // "Request" HTTP req library
3
4  const apiKey = 'f09a0c81d70542deb6e6bfb66d24c845' // api-key entregado por Banco Plaza
5  const apiSecret = 'bd18e09732874b04b4046f9c311426e3' // api-key-secret entregando por Banco Plaza
6
7  const apiPath = 'pagos/v0/p2p' // Path al recurso
8
9  const nonce = (Date.now() * 1000).toString() // Generación del nonce generator. Timestamp * 1000
10 const body = {} // Cuerpo de la petición JSON. Puede variar dependiendo del recurso que se esté invocando
11
12 let signature = `/${apiPath}${nonce}${JSON.stringify(body)}`
13 // Compuesto por el Path de la URL, el nonce, y el cuerpo de la petición
14
15 const sig = CryptoJS.HmacSHA384(signature, apiSecret).toString()
16 // Se firma la petición on la función HMAC usando la llave secreta o api-key-secret
17
18 const options = {
19   url: 'https://apiqa.bancoplaza.com:8585/${apiPath}',
20   headers: {
21     'api-key': apiKey,
22     'api-signature': sig,
23     'nonce': nonce
24   },
25   body: body,
26   json: true
27 }
28
29 request.post(options, (error, response, body) => {
30   console.log(body); // Se imprime la respuesta
31 })
32
```

Resumen de Operaciones

Cientes

Requiere autenticación	Endpoint	Método HTTP	Permisos
Sí	/v0/clientes/{id}	HEAD	READ
Sí	/v0/clientes/{id}	GET	READ

Recurso: /v0/clientes/{id}

Argumentos de Entrada

	Argumento	Tipo	Descripción
Argumentos Requeridos	id	String	<i>Identificación.</i> Se refiere al documento de identidad del cliente que se desea consultar. Debe incluir el tipo de persona. Por ejemplo, para un persona natural el valor sería V13759368. En contraste, para un comercio sería J00378944781. Longitud de doce (12) caracteres.
Argumentos Opcionales	telefono	String	<i>Teléfono.</i> Se refiere al número telefónico que el cliente posee registrado en la plataforma de Banco Plaza.
	email	String	E-mail. Se refiere al correo electrónico que el cliente posee registrado en la plataforma de Banco Plaza.
Permisos	READ		
Método HTTP	HEAD		
HTTP Status Code	204 No Content		
	400 Bad Request		
	401 Unauthorized		
	404 Not found		
	500 Internal Server Error		

Descripción

Permite verificar la existencia de un cliente Banco Plaza haciendo uso de su documento de identificación.

A continuación, un ejemplo de la petición:

GET /v0/clientes/V0016411999 HTTP/1.1

Request Header

Host: api.bancoplaza.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:76.0)

Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Api-key: bd18e09732874b04b4046f9c311426e3
Nonce: 758380027754
Api-signature:
b3e3742e9528bac62cbd1331a4976e29925e92d12361b49be6b09a8e33593f055152b2f1ef4100
05ff5fad49034b816e

Estructura de Salida

Se refiere a los datos que componen la estructura de salida para el recurso. Básicamente, es la respuesta a la petición. Se describe en la siguiente tabla:

Campo	Tipo	Descripción
codigoRespuesta	String	<i>Código de Respuesta.</i> Se refiere a un código de cuatro (4) dígitos que identifica la respuesta generada por el servicio. Este campo será agregado a la cabecera de la respuesta HTTP.
descripcionCliente	String	<i>Descripción para el Cliente.</i> Es la descripción asociada al <i>Código de Respuesta</i> diseñada para ser mostrada o desplegada a la capa front. Este campo será agregado a la cabecera de la respuesta HTTP.
descripcionSistema	String	<i>Descripción del Sistema.</i> Es la descripción asociada al <i>Código de Respuesta</i> diseñada para albergar información técnica sobre la respuesta a la petición. Este campo será agregado a la cabecera de la respuesta HTTP.
fechaHora	Datetime	<i>Fecha y Hora.</i> Es una marca de tiempo o timestamp del momento exacto en que se entrega la respuesta a la petición. Este campo será agregado a la cabecera de la respuesta HTTP.

A continuación, se muestra un ejemplo de una respuesta para este recurso:

Response Header

HTTP/1.1 204 No Content

codigoRespuesta: 0000

descripcionCliente: Transaccion Exitosa

descripcionSistema: Transaccion Exitosa

fechaHora: 2019-11-28 09:12:45

Códigos de Respuesta

La lista de los posibles Códigos de Respuesta se presenta a continuación:

Código de Respuesta	Descripción Sistema	Tipo de Resultado	HTTP Status Code
0000	TRANSACCIÓN EXITOSA	Transacción exitosa.	204
0002	PARÁMETRO [parametro] OBLIGATORIO	Validación de sistema	400
0003	PARÁMETRO [parametro] NO CUMPLE FORMATO	Validación de sistema	400
0010	CLIENTE NO EXISTE	Validación sistema de	404
0011	TELÉFONO NO COINCIDE	Validación sistema de	400
0012	EMAIL NO COINCIDE	Validación sistema de	400
9990	SERVICIO NO DISPONIBLE	Validación sistema de	400
9999	ERROR INTERNO	Error de sistema	500
A001	FIRMA DIGITAL INVÁLIDA	Validación sistema de	400
A002	FIRMA DIGITAL VENCIDA	Validación sistema de	400
A003	RECURSO NO AUTORIZADO	Validación sistema de	401
A004	API-KEY INVÁLIDA O REVOCADA	Validación sistema de	401

Recurso: /v0/clientes/{id}

Argumentos de Entrada

	Argumento	Tipo	Descripción
Argumentos Requeridos	id	String	<i>Identificación.</i> Se refiere al documento de identidad del cliente que se desea consultar. Debe incluir el tipo de persona. Por ejemplo, para persona natural el valor sería V13759368. En contraste, para un comercio sería J00378944781. Longitud de doce (12) caracteres.
Argumentos Opcionales	telefono	String	<i>Teléfono.</i> Se refiere al número telefónico que el cliente posee registrado en la plataforma de Banco Plaza.
	email	String	E-mail. Se refiere al correo electrónico que el cliente posee registrado en la plataforma de Banco Plaza.
Permisos	READ		
Método HTTP	GET		
HTTP Status Code			200 OK
			400 Bad Request
			401 Unauthorized
			404 Not found
			500 Internal Server Error

Descripción

Permite recuperar la información de un cliente a partir del documento de identificación indicado en {id}.

A continuación, un ejemplo de la petición:

GET /v0/clientes/V0016411999 HTTP/1.1

Request Header

Host: api.bancoplaza.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:76.0)

Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Api-key: bd18e09732874b04b4046f9c311426e3
Nonce: 758380027754
Api-signature:
b3e3742e9528bac62cbd1331a4976e29925e92d12361b49be6b09a8e33593f055152b2f1ef4100
05ff5fad49034b816e

Estructura de Salida

Se refiere a los datos que componen la estructura de salida para el recurso. Básicamente, es la respuesta a la petición. Se describe en la siguiente tabla:

Campo	Tipo	Descripción
codigoRespuesta	String	<i>Código de Respuesta.</i> Se refiere a un código de cuatro (4) dígitos que identifica la respuesta generada por el servicio. Este campo será agregado a la cabecera de la respuesta HTTP.
descripcionCliente	String	<i>Descripción para el Cliente.</i> Es la descripción asociada al <i>Código de Respuesta</i> diseñada para ser mostrada o desplegada a la capa front. Este campo será agregado a la cabecera de la respuesta HTTP.
descripcionSistema	String	<i>Descripción del Sistema.</i> Es la descripción asociada al <i>Código de Respuesta</i> diseñada para albergar información técnica sobre la respuesta a la petición. Este campo será agregado a la cabecera de la respuesta HTTP.
fechaHora	Datetime	<i>Fecha y Hora.</i> Es una marca de tiempo o timestamp del momento exacto en que se entrega la respuesta a la petición. Este campo será agregado a la cabecera de la respuesta HTTP.
id	String	<i>Identificación.</i> Se refiere al documento de identidad del cliente que fue indicado como argumento de entrada.

nombreCompleto	String	<i>Nombre Completo.</i> Se refiere a los nombres y a apellidos del cliente consultado.
----------------	--------	--

Un ejemplo de cómo luce una respuesta es el siguiente:

Response Header

HTTP/1.1 200 OK

codigoRespuesta: 0000

descripcionCliente: Transaccion Exitosa

descripcionSistema: Transaccion Exitosa

fechaHora: 2019-11-28 09:12:45

Body

```
{
    "id": "V0016411999"
    , "nombreCompleto": "JAVIER ENRIQUE, MARAVER SALAS"
}
```

Códigos de Respuesta

Los códigos de respuesta son exactamente los mismos que los que se presentan en el recurso </v0/clientes/{id}>, con la salvedad que el código de estatus HTTP para las respuestas exitosas es 200 y no 204.