BANCO PLAZA

Tú cuentas

Referencia de Uso: API Banco Plaza

Comprobación de Pagos P2P/P2C

Mayo 2026

Versión 4.0.0

Tabla de Contenido

Datos de Dominio

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

URL para consumo API

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	openapiqa.bancoplaza.com	N/A
Productivo	HTTPS	openapi.bancoplaza.com	N/A

URL para generación de token de autenticación API

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	portalapiqa.bancoplaza.com/oauth2/token	N/A
Productivo	HTTPS	portalapi.bancoplaza.com/oauth2/token	N/A

Procedimiento para la autenticación de APIs con Oauth 2.0

OAuth 2.0 es un marco de autorización (Authorization Framework) basado en principios arquitectónicos y estándares de seguridad globales. Su objetivo principal es delegar el acceso a recursos protegidos sin la necesidad de compartir las credenciales directas del usuario. En el ecosistema de WSO2, el endpoint /oauth2/token actúa como la puerta central para la gestión de acceso. Su función es realizar el intercambio de credenciales de autorización (como códigos de autorización o credenciales de aplicación) por un Access Token válido.

Requisitos Obligatorios de la Petición

Para garantizar la integridad y seguridad del proceso, todas las solicitudes al endpoint /oauth2/token deben contemplar los valores del `client_id` y `client_secret` emitidos por el WSO2 APIM

Definición: CLIENT_ID

Es el identificador único que permite al servidor reconocer qué aplicación específica está solicitando el acceso.
Ejemplo: client_id

VgV9jtzb6ozPr_mqwlckJFR5C1oa

Definición: CLIENT_SECRET

Es la clave secreta o contraseña de la aplicación que verifica su identidad y asegura que solo entidades autorizadas puedan interactuar con la API.
Ejemplo: client_secret

FJ2lciJ3xXW7LhFa2f7oZ_HKNZsa

Definición: URL TOKEN

Es el recurso de red (punto de acceso) dedicado exclusivamente a la emisión de tokens de acceso. En el modelo de OAuth 2.0, este endpoint actúa como el servidor de autorización.

Características Principales:

- Método Obligatorio: Es estrictamente de tipo POST. Esto es así porque el cuerpo de la petición contiene credenciales sensibles y porque la operación no es "idempotente" (estás creando una sesión o recurso nuevo).
- Seguridad (SSL/TLS): Al manejar credenciales, este endpoint solo debe estar disponible a través de HTTPS.
- Procesamiento: Cuando la URL recibe la petición, el Key Manager de WSO2 realiza tres pasos internos:
- Valida la autenticidad del client_id y client_secret.
- Verifica que la aplicación esté activa.
- Genera un token firmado con un tiempo de expiración definido.

Procedimiento para la solicitud de token

Antes de consumir el endpoint, necesitas las credenciales emitidas por Banco Plaza.

- client_id: Tu identificador único
- client_secret: Tu contraseña de aplicación.

La solicitud se envía mediante un método POST a la URL del consumo del token. Este paso es el intercambio formal de identidad por acceso a continuación se describen la información de acuerdo con el ambiente.

URL de consumo del token

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	portalapiqa.bancoplaza.com/oauth2/token	N/A
Productivo	HTTPS	portalapi.bancoplaza.com/oauth2/token	N/A

Para realizar el proceso de consumo de las APIs se requiere la emisión del token, para ello se ha incorporado un script basado en el concepto de encadenamiento de peticiones, su funcionalidad es obtener un token y posteriormente enviarlo como parámetro al header: Authorization.

A continuación, se mencionan los parámetros a considerar remplazar su valor para consumo de las APIS en un código de ejemplo en Javascript y la explicación de este

Los valores que se deben de cambiar son:

tokenUrl = Url del token

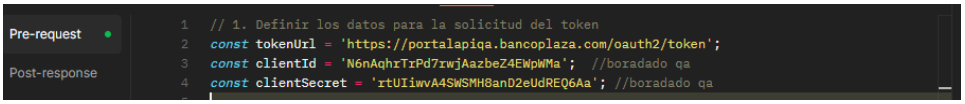
Ejemplo: https://portalapiqa.bancoplaza.com/oauth2/token

clientId = valor del client_id

Ejemplo: N6nAqhrTrPd7rwjAazbeZ4EWpWMa

clientSecret = valor del client_secret

Ejemplo: rtUIiwvA4SWSMH8anD2eUdREQ6Aa



Nota: cada uno de los recursos se deberá hacer el remplazo de la información que anteriormente se menciona

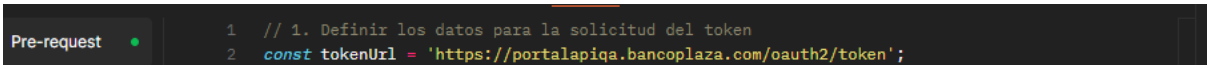
La función principal del script es mostrar el proceso a seguir para automatizar la obtención de un token de acceso (OAuth 2.0) antes de ejecutar la petición principal de la colección.

A continuación, se muestra el desglose paso a paso:

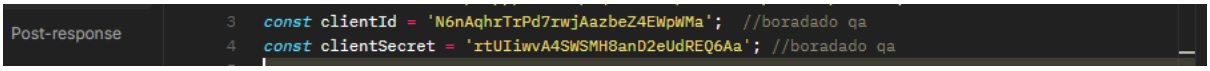
1. Configuración de Credenciales

   Se definen las variables necesarias para identificarse ante el servidor de Banco Plaza:

- **URL del Token:** La dirección del servidor de autorización (tokenUrl).



- **Credenciales:** Define el clientId y el clientSecret.

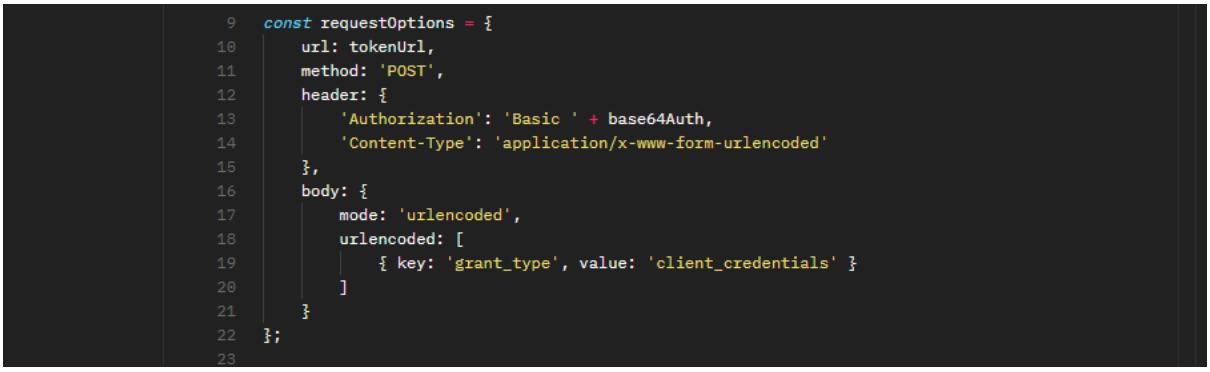


- **Codificación:** Convierte estas credenciales a un formato **Base64** usando la función btoa(). Esto es un requisito estándar para el encabezado de "Authorization: Basic".

2. Definición del Objeto de Petición (requestOptions)

Se construye la estructura de la solicitud HTTP que se enviará:

- Método: POST.
- Headers: Configura el tipo de contenido como application/x-www-form-urlencoded e incluye el token de autorización básica que se generó en el paso anterior.
- Cuerpo (Body): Define que el flujo de autenticación es de tipo client_credentials, que es el usado para comunicación de servidor a servidor (sin intervención de un usuario final).



3. Ejecución de la Petición (pm.sendRequest)

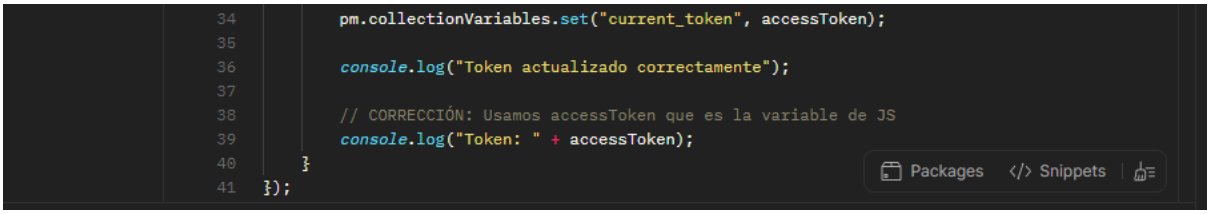
El script envía la solicitud de forma asíncrona:

- Manejo de Errores: Si algo sale mal (problemas de red, URL incorrecta), imprime el error en la consola.
- Procesamiento de Respuesta: Si la petición es exitosa, convierte la respuesta del servidor en un objeto JSON para poder leerla.

4. Extracción y Almacenamiento del Token

Esta es la parte más importante para el flujo de trabajo:

- Extracción: Toma el valor del campo access_token que devolvió el banco.
- Persistencia: Guarda ese valor en una variable de colección llamada current_token.
- Log: Imprime el token en la consola para que el desarrollador pueda verificar que se recibió correctamente.



Resumen de Operaciones

Pagos P2P/P2C

Requiere autenticación	Endpoint	Método HTTP	Permisos
Sí	/pagos/p2p/v1/v1/pagos/p2p/{id}	GET	READ

Recurso: `/pagos/p2p/v1/v1/pagos/p2p/{id}`

Argumentos de Entrada

A continuación, se listan los argumentos de entrada que se utilizarán para definir criterios de búsqueda especializados. Todos los argumentos de entradas deben ser indicados en el `QueryString`.

	Argumento	Tipo	Descripción
Argumentos Requeridos	`id`	`String`	*Identificación*. Se refiere al documento de identidad del cliente del que se desea consultar la información. 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.
	`canal`	`String`	*Canal*. Código que busca clasificar el origen de la transacción. Los posibles valores son los siguientes: ● "20" para POS ● "21" para MERCHANT ● "22" para VPOS ● "23" para BOTON DE PAGO ● “24” para BILLETERA DIGITAL Entre otros
Argumentos Opcionales	`acc`	`Integer`	*Acción*. Se utiliza para indicar la dirección de las transacciones P2P/P2C a consultar. ● Cero (`0`) para transacciones entrantes. ● Uno (`1`) para transacciones salientes. ● Mayor que uno (`1`) para todas las transacciones. En el caso de que el argumento esté ausente, se asume que su valor es igual a cero (`0`).
	`fi`	`String`	*Fecha Inicio*. Se utiliza para filtrar por todas aquellas transacciones P2P/P2C cuya fecha sea igual o mayor al valor indicado en el argumento. El formato para el argumento es el ISO 8601. Ej.: `YYYY-MM-DD`.

		Si el valor del argumento se encuentra por debajo del límite definido para el servicio P2P/P2C, entonces `fi` toma el valor establecido por el límite. El valor del límite del servicio es de seis (`2`) meses, pero puede variar sin previo aviso. Este argumento se puede usar junto con el argumento `ff` para definir filtros de rangos de fechas. En caso de que el argumento esté ausente, asume el valor del límite del servicio, salvo que el argumento ff también se encuentre ausente, en cuyo caso asumirá el valor de la fecha corriente.
`ff`	`String`	*Fecha Fin*. Se utiliza para filtrar por todas aquellas transacciones P2P/P2C cuya fecha sea igual o menor al valor indicado en el argumento. El formato para el argumento es el `ISO 8601`. Ej.: `YYYY-MM-DD`. Si el valor del argumento es mayor a la fecha corriente, entonces ff tomará el valor de la fecha corriente. Este argumento se puede usar junto con el argumento `fi` para definir filtros de rangos de fechas. El en caso de que el argumento esté ausente, asume el valor de la fecha corriente.
`tlf`	`String`	*Teléfono*. Se utiliza para filtrar por el número telefónico dadas las siguientes condiciones: ● Si el argumento `acc` es cero (`0`), entonces retorna todas aquellas transacciones recibidas por el número telefónico `tlf`. ● Si el argumento `acc` es uno (`1`), entonces retorna todas aquellas transacciones enviadas al número telefónico `tlf`. ● Si el argumento `acc` es mayor a uno (`1`), entonces retorna todas aquellas transacciones recibidas por y enviadas al número telefónico `tlf`.

		En el caso de que el argumento esté ausente, no se asume ningún valor por omisión, simplemente es ignorado como filtro. El formato que debe cumplir el parámetro de teléfono es como sigue: [código de país] + [Código Operadora] + [Número Teléfono]. Ejemplo: 00582122578567. No debe llegar ningún carácter especial, tal como paréntesis, guiones o puntos.
`tlfa`	`String`	*Teléfono Afiliado.* Se refiere al número de teléfono asociado a la afiliación del propietario de las llaves. En el caso de que el argumento esté ausente, se asume el valor por omisión, que es el informado al equipo de Banco Plaza al momento de suscribirse al servicio de Open Bank. El formato que debe cumplir el parámetro de teléfono es como sigue: [código de país] + [Código Operadora] + [Número Teléfono]. Ejemplo: 00582122578567. No debe llevar ningún carácter especial, tal como paréntesis, guiones o puntos.
`horaIni`	`String`	*Hora Inicio*. Se utiliza para filtrar por todas aquellas transacciones P2P/P2C cuya hora sea igual o mayor al valor indicado en el argumento. El formato para el argumento es tiempo militar HHMMSSMS. Ejemplos: 15350501. En tiempo militar, las horas se representan de 00 a 23, donde 00:00 (medianoche) es el inicio del día y 23:59 (11:59) es el último minuto del día. Este argumento se puede usar junto con el argumento “`horaFin`” para definir filtros de rangos de tiempo. En caso de que el argumento esté ausente, asume el valor del límite del servicio, salvo que el argumento “`horaFin`” también se encuentre

			ausente, en cuyo caso asumirá el valor para el día completo.
	horaFin	String	Hora Fin. Se utiliza para filtrar por todas aquellas transacciones P2P/P2C cuya hora sea igual o menor al valor indicado en el argumento. El formato para el argumento es tiempo militar HHMMSSMS. Ejemplos: 15350501. En tiempo militar, las horas se representan de 00 a 23, donde 00:00 (medianoche) es el inicio del día y 23:59 (11:59) es el último minuto del día. Este argumento se puede usar junto con el argumento “horaIni” para definir filtros de rangos de tiempo. El en caso de que el argumento esté ausente, asume el valor de la hora corriente.
	id_Pago	String	Identificación del cliente que paga o recibe Si la acción es un pago recibido, se refiere a la identificación de quien envía o hace el pago. Si la acción es un pago enviado, se refiere a la identificación de quien recibe el pago. Por ejemplo, para persona natural el valor sería `V00013759368`. En contraste, para un comercio sería `J00378944781`. Longitud de doce (`12`) caracteres.
Permisos	`READ`		
Método HTTP	`GET`		

HTTP Status Code			`200 Accepted`
			`204 No Content`
			`400 Bad Request`
			`401 Unauthorized`
			`500 Internal Server Error`

Descripción

Permite realizar consultas sobre el historial de transacciones P2P/P2C del cliente que invoca el recurso, haciendo uso de los distintos argumentos o filtros dispuestos para tal fin. La identificación del cliente se determina a través del client_id y el client_secret.

Algunos ejemplos de uso podrían ser los siguientes:

QueryString	**Resultado Esperado**
`Omitida`	Retorna las transacciones recibidas para el día de hoy.
`?tlf=[valor]`	Retorna las transacciones recibidas para el día de hoy originadas por el número de teléfono especificado en `[valor]`.
`?tlf=[valor]&acc=1`	Retorna las transacciones enviadas al número de teléfono especificado en `[valor]` el día de hoy.
`?fi=[value]&ff=[value]`	Retorna las transacciones recibidas en el rango de fecha especificado para `fi` y `ff`.
`? horaIni=[value]&horaFin=[value]`	Retorna las transacciones recibidas en el rango de horas especificado para `fi` y `ff`.
`? id_Pago=[value]`	Pagos recibidos: Retorna las transacciones recibidas cuyo ordenante coincida con identificación del id_Pago. Pagos enviados: Retorna las transacciones enviadas cuyo beneficiario coincida con identificación del id_Pago.

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`	*Código de Respuesta*. 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`	*Descripción para el Cliente*. Es la descripción asociada al *Código de Respuesta* diseñada para ser mostrada o desplegada a la capa front.

		Este campo será agregado a la cabecera de la respuesta HTTP.
descripcionSistema	String	*Descripción del Sistema*. Es la descripción asociada al Código de Respuesta diseñada para albergar información técnica sobre la respuesta a la petición. **Este campo será agregado a la cabecera de la respuesta HTTP.**
fechaHora	Datetime	*Fecha y Hora*. 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.**
pagos	List<Pago>	*Lista de Pagos*. Se refiere a una lista de objetos JSON que contiene la información de cada uno de los pagos que cumplieron con el criterio de búsqueda suministrado.
cantidadPagos	Integer	*Cantidad de Pagos*. Se refiere a la cantidad de objetos Pago dentro de la lista pagos. **Este campo será agregado a la cabecera de la respuesta HTTP.**

Dado que pagos es una lista de objetos Pago, es necesario detallar la estructura de dicho objeto. En la siguiente tabla se listan los atributos que lo componen:

Campo	Tipo	Descripción
accion	String	*Acción.* Indica si se trata de un pago recibido (R) o enviado (P).
banco	String	*Banco.* Código de cuatro (4) dígitos que identifica a la entidad bancaria ordenante de la transacción si acción es igual a "R" o a la beneficiaria si acción es igual a "P".
telefonoCliente	String	*Teléfono Cliente.* Numero de teléfono del ordenante de la transacción si acción es igual a "R" o del beneficiario si acción es "P".
telefonoAfiliado	String	*Teléfono Afiliado.* Número de teléfono del cliente afiliado al servicio de P2P/P2C en Banco Plaza. Es

		indistinto del valor de `acción`.
`monto`	`Double`	*Monto.* Se refiere al valor de la transacción P2P expresado en bolívares.
`fecha`	`String`	*Fecha.* Se trata de la fecha de la transacción en el siguiente formato: YYYYMMDD.
`hora`	`String`	*Hora. Se trata de la hora de la transacción en el siguiente formato: HHMM*
`referencia`	`String`	*Número de referencia.* Código generado por el sistema para identificar de forma unívoca el pago dentro del sistema interbancario.
`concepto`	`String`	*Concepto.* Descripción que el cliente le da la transacción.
`cedulaB`	`String`	Identificación del cliente que paga o recibe Si la acción es “R”, se refiere a la identificación de quien envía o hace el pago. Si la acción es “P”, se refiere a la identificación de quien recibe el pago.

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

Response Header

```
HTTP/1.1 200 OK

codigoRespuesta: 0000
descripcionCliente: Transaccion Exitosa
descripcionSistema: Transaccion Exitosa
fechaHora: 2019-11-28 09:12:45
```

Body

```
{
          cantidadPagos: 3
        , "pagos": [
            {
                  "accion": "R",
                  "banco": "0172",
                  "concepto": "pago",
                  "fecha": "20260412",
                  "hora": "18325724",
                  "monto": "500.00",
```

```
                "referencia": "183255935841",
                "telefonoAfiliado": "4242956418",
                "telefonoCliente": "4124688296",
                "cedulaB ": "V00025709325"

            },
            {
                "accion": "R",
                "banco": "0007",
                "concepto": "P2C BCO DIGITAL D LOS TRABAJADORES ",
                "fecha": "20260513",
                "hora": "17515985",
                "monto": "7000.00",
                "referencia": "009281129281",
                "telefonoAfiliado": "4242956488",
                "telefonoCliente": "4145951774",
                "cedulaB ": "J00500709325"

            },
                "accion": "R",
                "banco": "0102",
                "concepto": "Pagom vilBDV                          ",
                "fecha": "20260515",
                "hora": "17191778",
                "monto": "600.00",
                "referencia": "000779144436",
                "telefonoAfiliado": "4242956418",
                "telefonoCliente": "4125517941"
                "cedulaB ": "V00500709995"

            }
        ]
}
```


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.	200
0002	PARÁMETRO [parametro] OBLIGATORIO	Error de sistema.	400
E001	CLIENTE NO REGISTRADO	Validación de sistema	500

E002	TELEFONO NO REGISTRADO	Validación de sistema	500
E003	REGISTRO NO EXISTE	Validación de sistema	204
E024	CLIENTE BLOQUEADO	Validación de sistema	500
N002	CEDULA/RIF NO NUMERICA	Validación de sistema	400
N003	TELEFONO NO NUMERICO	Validación de sistema	400
N004	TELEFONO 1 NO NUMERICO	Validación de sistema	400
N005	TELEFONO 2 NO NUMERICO	Validación de sistema	400
V008	TELEFONO NO VALIDO	Validación de sistema	400
0096	ERROR EN SISTEMA	Error de sistema	500
A001	FIRMA DIGITAL INVÁLIDA	Error de sistema	400
A002	FIRMA DIGITAL VENCIDA	Error de sistema	500
A003	RECURSO NO AUTORIZADO	Validación de sistema	401
A004	API-KEY INVÁLIDA O REVOCADA	Validación de sistema	401
0095	HORA INICIO INVALIDA	Error de sistema.	400
0094	HORA FIN INVALIDA	Error de sistema.	400